Enterprise-wide Optimization for Industrial Demand Side Management

Qi Zhang
Carnegie Mellon University

Follow this and additional works at: http://repository.cmu.edu/dissertations

Recommended Citation
Zhang, Qi, "Enterprise-wide Optimization for Industrial Demand Side Management" (2016). Dissertations. 823.
http://repository.cmu.edu/dissertations/823

This Dissertation is brought to you for free and open access by the Theses and Dissertations at Research Showcase @ CMU. It has been accepted for inclusion in Dissertations by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.
Enterprise-wide Optimization for Industrial Demand Side Management

Qi Zhang

Chemical Engineering

4/15/16

4/15/16

4/15/16
ENTERPRISE-WIDE OPTIMIZATION FOR INDUSTRIAL DEMAND SIDE MANAGEMENT

Submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY in
CHEMICAL ENGINEERING

QI ZHANG
M.Sc., Advanced Chemical Engineering, Imperial College London
B.Sc., Mechanical Engineering, RWTH Aachen University

Carnegie Mellon University
Pittsburgh, Pennsylvania

April 2016
To my family,
and in memory of Grandpa.
ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to my thesis advisor, Prof. Ignacio E. Grossmann, who has been a mentor and role model to me for longer than just my Ph.D. years. I met Ignacio the first time in 2009; back then, I was an undergraduate exchange student at Carnegie Mellon University, young and clueless about many things in life. It was Ignacio who introduced me to process systems engineering and later provided me with the unique opportunity to pursue my research interests in his group. Ever since, I have greatly benefited from his knowledge, guidance, encouragement, and the unprecedented academic freedom that he gave me, which I highly appreciated. Ignacio’s passion for academic excellence and his high ethical standards have profoundly influenced both my academic and personal development. I am honored to have received his generosity and care, for which I will be forever grateful.

I am very much indebted to our collaborators from Praxair, in particular Dr. Jose M. Pinto and Dr. Arul Sundaramoorthy. It is certainly not an overstatement to say that Jose has been like a second advisor to me. I am grateful for his mentorship (at both the academic and personal level) and all his insightful feedback that has guided me through my Ph.D. work. I thank Arul for his patience and the many hours he has spent working with me, especially during my longer stays at the Praxair Technology Center. It has been a pleasure to work with such supportive industrial collaborators, whose input has significantly increased the quality and practical relevance of the work.

I would like to thank my dear academic brother Dr. Sumit Mitra, from whom I inherited the project on industrial demand side management. Sumit’s previous contributions have laid the foundation for this thesis, and I have greatly benefited from his advice and insights. Many thanks also go to Dr. Ricardo M. Lima for inspiring research discussions, which ultimately led to an excellent collaboration that I very much enjoyed.

During my time at Carnegie Mellon, I had the privilege to supervise the fol-
ollowing very talented undergraduate and master’s students in their research work: Luise F. Bering, Andreas M. Bremen, Jochen L. Cremer, Lorenz H. J. Fleitmann, Caroline Ganzer, Clara F. Heuberger, and Michael F. Morari. Although the works of some of them are not included in this thesis, I want to thank them all for their hard work, their willingness to take the chance with me, and the many invaluable lessons they taught me about how to be an effective advisor and mentor.

My thanks further go to the members of my thesis committee: Prof. Lorenz T. Biegler, Prof. Antonio J. Conejo, Prof. Chrysanthos E. Gounaris, Prof. Jeremy J. Michalek, and Dr. Jose M. Pinto. I thank them for their valuable feedback and the effort they have spent on critically evaluating this work.

I am grateful for the financial support from Praxair and from the National Science Foundation under Grant No. 1159443.

With great appreciation, I thank my friends here in the U.S., in Germany, in China, and elsewhere in the world for all the fantastic memories and for being there whenever I needed them. Finally, I want to thank my grandparents, my parents, and my sister for their unconditional love and support. Special thanks to Grandpa, who has taught me more than I first realized, and more than I had time to tell him.
ABSTRACT

In the light of increasing volatility in electricity price and availability, demand side management (DSM), which refers to the active management of electricity consumption, has become crucial for the economic performance of power-intensive industries. Due to its time-sensitive nature, DSM is a challenge for industrial plants; however, it can also be an opportunity if sufficient process flexibility is available, which can be leveraged to take advantage of financial incentives provided by various electricity markets.

The goal of this work is to develop systematic decision-making tools for industrial DSM at the enterprise level. We identify and address four major challenges: (1) accurate modeling of operational flexibility, (2) integration of production and energy management, (3) decision-making across multiple time scales, and (4) optimization under uncertainty.

We develop a discrete-time mixed-integer linear programming (MILP) model that integrates detailed production scheduling and electricity procurement from various sources. The proposed model is proven to be computationally efficient, which can be attributed to the mode-based formulation and the incorporation of a special type of process surrogate model, referred to as Convex Region Surrogate (CRS). In a CRS model, the feasible region is given by the union of convex regions in the form of polytopes, and for each region, the corresponding cost function is approximated by a linear function. For the construction of CRS models, we propose a data-driven algorithm that can be applied to data obtained from the real process or from simulations.

Using the proposed integrated scheduling model as a basis, we optimize decisions regarding load shifting, inventory management, electricity procurement, energy storage, provision of interruptible load, etc. The framework is further extended to the supply chain level by also integrating distribution decisions, resulting in a multiscale production routing problem (MPRP). In order to solve large instances of the MPRP, we propose an iterative MILP-based heuristic algorithm
that obtains high-quality solutions in reasonable computation times.

A strong focus of this work lies on the treatment of uncertainty, which occurs in many forms in industrial DSM problems. We consider uncertainty in product demand, electricity price, and grid contingency events. These uncertainties all have different characteristics and affect the process in different ways; hence, we consider them using different modeling strategies, namely stochastic programming and robust optimization. We emphasize the consideration of risk, which is incorporated into the stochastic programming model using the conditional value-at-risk. In the proposed robust optimization models, we reduce the level of conservatism by implementing appropriate budget uncertainty sets and incorporating recourse decisions in the form of linear decision rules. Computational challenges are addressed by applying reformulations and decomposition strategies. Finally, we examine for linear systems the relationship between flexibility analysis and robust optimization.

The effectiveness of the proposed methodologies is demonstrated in several case studies, many of which consider industrial test cases with real-world data provided by Praxair.
CONTENTS

List of Figures XVII
List of Tables XX

1. Introduction 1
  1.1. Challenges in the Modern Power Grid .............................. 2
  1.2. Smart Grid and Demand Side Management (DSM) .................. 4
  1.3. Definition and Classification of DSM Activities .................. 6
  1.4. Characteristics of Industrial DSM ................................. 8
  1.5. Advances and Challenges in Planning and Scheduling for
      Industrial DSM ....................................................... 10
     1.5.1. Modeling Operational Flexibility ............................. 12
     1.5.2. Integration of Production and Energy Management .......... 18
     1.5.3. Decision-making Across Multiple Time Scales ................. 19
     1.5.4. Optimization Under Uncertainty ............................. 19
  1.6. Outline of the Thesis ............................................. 22

2. Data-driven Construction of Convex Region Surrogate Models 26
  2.1. Background ....................................................... 27
  2.2. Convex Region Surrogate .......................................... 29
  2.3. Formal Problem Statement ........................................ 33
  2.4. Illustrative Overview of the Algorithm ........................... 34
  2.5. Phase 1: Subset Assignment ....................................... 35
     2.5.1. Subset Assignment Formulation ............................... 35
     2.5.2. Identifying Overlapping Convex Hulls ....................... 38
     2.5.3. Overlap Elimination Cuts .................................. 39
  2.6. Phase 2: Construction of Convex Regions ......................... 40
     2.6.1. Detecting Contour of the Feasible Region ................. 42
     2.6.2. Constructing Convex Regions ............................... 50
## CONTENTS

2.7. Summary of the Algorithm ........................................... 56
2.8. Numerical Results .................................................. 58
   2.8.1. Computational Study ......................................... 59
   2.8.2. Industrial Case Study ....................................... 67
2.9. Remarks .............................................................. 68
2.10. Summary ............................................................. 70

   3.1. Background ......................................................... 72
   3.2. Problem Statement .............................................. 73
   3.3. Model Formulation ............................................... 74
      3.3.1. Time Representation ....................................... 74
      3.3.2. Process Network Representation ......................... 74
      3.3.3. Mass Balance Constraints ................................. 75
      3.3.4. Process Surrogate Model ................................ 76
      3.3.5. Transition Constraints .................................. 79
      3.3.6. Energy Balance Constraints .............................. 80
      3.3.7. Power Contract Model .................................. 80
      3.3.8. Boundary Conditions ..................................... 84
      3.3.9. Objective Function ....................................... 84
   3.4. Numerical Results ............................................... 85
      3.4.1. Illustrative Example ....................................... 85
      3.4.2. Industrial Case Study ................................... 93
   3.5. Summary ............................................................. 96

4. Risk-based Integrated Production Scheduling and Electricity Procurement 98
   4.1. Background ......................................................... 98
   4.2. Problem Statement .............................................. 100
   4.3. Model Formulation ............................................... 101
      4.3.1. Uncertainty Modeling Strategy ......................... 101
      4.3.2. Plant Model ................................................ 102
      4.3.3. Transition Constraints .................................. 103
      4.3.4. Mass Balance Constraints ................................. 103
      4.3.5. Energy Balance Constraints .............................. 104
      4.3.6. Power Contract Model .................................. 104
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.7.</td>
<td>Boundary Conditions</td>
<td>106</td>
</tr>
<tr>
<td>4.3.8.</td>
<td>Total Expected Operating Cost</td>
<td>107</td>
</tr>
<tr>
<td>4.3.9.</td>
<td>Conditional Value-at-Risk</td>
<td>107</td>
</tr>
<tr>
<td>4.3.10.</td>
<td>Objective Functions</td>
<td>109</td>
</tr>
<tr>
<td>4.4.</td>
<td>Scenario Generation and Reduction</td>
<td>109</td>
</tr>
<tr>
<td>4.4.1.</td>
<td>Scenario Generation Using ARIMA/ARIMAX Models</td>
<td>110</td>
</tr>
<tr>
<td>4.4.2.</td>
<td>Scenario Reduction Using Probability Distance Metrics</td>
<td>111</td>
</tr>
<tr>
<td>4.5.</td>
<td>Multicut Benders Decomposition</td>
<td>112</td>
</tr>
<tr>
<td>4.6.</td>
<td>Numerical Results</td>
<td>114</td>
</tr>
<tr>
<td>4.6.1.</td>
<td>Illustrative Example</td>
<td>114</td>
</tr>
<tr>
<td>4.6.2.</td>
<td>Industrial Case Study</td>
<td>123</td>
</tr>
<tr>
<td>4.6.3.</td>
<td>Discussion</td>
<td>130</td>
</tr>
<tr>
<td>4.7.</td>
<td>Summary</td>
<td>132</td>
</tr>
<tr>
<td>5.</td>
<td>Robust Scheduling of Air Separation Plants with Cryogenic Energy</td>
<td></td>
</tr>
<tr>
<td>5.1.</td>
<td>Background</td>
<td>134</td>
</tr>
<tr>
<td>5.2.</td>
<td>Problem Statement</td>
<td>137</td>
</tr>
<tr>
<td>5.3.</td>
<td>Model Formulation</td>
<td>138</td>
</tr>
<tr>
<td>5.3.1.</td>
<td>Mass Balance Constraints</td>
<td>138</td>
</tr>
<tr>
<td>5.3.2.</td>
<td>Energy Balance Constraints</td>
<td>140</td>
</tr>
<tr>
<td>5.3.3.</td>
<td>ASU Scheduling Model</td>
<td>141</td>
</tr>
<tr>
<td>5.3.4.</td>
<td>Boundary Conditions</td>
<td>142</td>
</tr>
<tr>
<td>5.3.5.</td>
<td>Objective Function</td>
<td>142</td>
</tr>
<tr>
<td>5.4.</td>
<td>Robust Model with Reserve Market Participation</td>
<td>143</td>
</tr>
<tr>
<td>5.4.1.</td>
<td>Uncertainty Set</td>
<td>144</td>
</tr>
<tr>
<td>5.4.2.</td>
<td>Robust Counterpart</td>
<td>146</td>
</tr>
<tr>
<td>5.5.</td>
<td>Industrial Case Study</td>
<td>149</td>
</tr>
<tr>
<td>5.5.1.</td>
<td>ASU-CES with Energy Market Participation</td>
<td>150</td>
</tr>
<tr>
<td>5.5.2.</td>
<td>ASU-CES with Energy and Reserve Market Participation</td>
<td>151</td>
</tr>
<tr>
<td>5.5.3.</td>
<td>Sensitivity Analysis</td>
<td>154</td>
</tr>
<tr>
<td>5.5.4.</td>
<td>Computational Results</td>
<td>156</td>
</tr>
<tr>
<td>5.6.</td>
<td>Summary</td>
<td>156</td>
</tr>
<tr>
<td>6.</td>
<td>Adjustable Robust Scheduling of Continuous Industrial Processes</td>
<td></td>
</tr>
<tr>
<td>6.1.</td>
<td>Background</td>
<td>157</td>
</tr>
</tbody>
</table>
CONTENTS

6.2. Problem Statement ................................................. 159
6.3. Uncertain Scheduling Model ..................................... 160
  6.3.1. Production Scheduling Model ............................... 161
  6.3.2. Interruptible Load Constraints ............................. 162
  6.3.3. Objective Function ......................................... 163
  6.3.4. Uncertain Optimization Problem ......................... 163
  6.3.5. Uncertainty Set ........................................... 164
6.4. Adjustable Robust Counterpart ................................. 165
  6.4.1. Linear Decision Rules ..................................... 165
  6.4.2. Reformulation of the Plant Model ....................... 166
  6.4.3. Elimination of State Variables ............................ 167
  6.4.4. Linearly Adjustable Robust Counterpart ................ 168
  6.4.5. Remark on Robust Formulation Without Recourse ....... 170
  6.4.6. Tightening Constraints ................................... 171
6.5. Numerical Results ................................................ 171
  6.5.1. Illustrative Example ....................................... 172
  6.5.2. Industrial Case Study ..................................... 178
6.6. Summary .......................................................... 181

7. Multiscale Production Routing in Supply Chains with Power-intensive Production Facilities ................................................. 184
  7.1. Background ....................................................... 185
  7.2. Problem Statement .............................................. 189
  7.3. Model Formulation .............................................. 190
    7.3.1. Multiscale Time Representation ......................... 190
    7.3.2. Production Scheduling ..................................... 191
    7.3.3. Distribution Planning ................................... 194
    7.3.4. Objective Function ....................................... 196
  7.4. Solution Method ................................................ 197
    7.4.1. Initialization ............................................... 198
    7.4.2. Updating Set of Candidate Routes ..................... 199
    7.4.3. Stopping Criteria .......................................... 202
    7.4.4. Algorithmic Parameters ................................. 202
  7.5. Numerical Results .............................................. 203
    7.5.1. Illustrative Example ....................................... 204
    7.5.2. Computational Study ..................................... 209
## CONTENTS

7.5.3. Industrial Case Study .............................................. 216
7.6. Summary .............................................................. 221

8. On the Relation Between Flexibility Analysis and Robust Optimization for Linear Systems ................. 223
  8.1. Background ......................................................... 224
  8.2. Historical Perspective ............................................. 226
  8.3. Flexibility Test Problem ........................................... 230
    8.3.1. Problem Statement ........................................... 230
    8.3.2. Traditional Flexibility Analysis ............................... 231
    8.3.3. Duality-based Flexibility Analysis ........................... 233
    8.3.4. Affinely Adjustable Robust Optimization .................... 235
  8.4. Flexibility Index Problem .......................................... 241
    8.4.1. Problem Statement ........................................... 241
    8.4.2. Traditional Flexibility Analysis ............................... 241
    8.4.3. Duality-based Flexibility Analysis ........................... 242
    8.4.4. Affinely Adjustable Robust Optimization .................... 243
  8.5. Design Under Uncertainty with Flexibility Constraints ......................................................... 244
    8.5.1. Problem Statement ........................................... 244
    8.5.2. Flexibility Analysis .......................................... 245
    8.5.3. Affinely Adjustable Robust Optimization .................... 246
  8.6. Numerical Results .................................................. 247
    8.6.1. Example 1: Heat Exchanger Network .......................... 247
    8.6.2. Example 2: Process Flowsheet ................................ 250
    8.6.3. Example 3: Planning of a Large-Scale Process Network .... 252
  8.7. Summary .............................................................. 257

9. Conclusions .............................................................. 259
  9.1. Summary of the Thesis ............................................. 259
  9.2. Research Contributions ........................................... 265
  9.3. Directions for Future Work ....................................... 268
    9.3.1. Construction and Validation of CRS Models .................. 268
    9.3.2. Industrial DSM Extensions .................................... 269
    9.3.3. Multiscale Integrated Optimization .......................... 271
    9.3.4. Optimization Under Uncertainty ............................... 272
    9.3.5. Extension to Nonlinear Models ................................. 273
A. Alternative Formulation for CRS Model 275
B. Data for CRS Illustrative Example 277
C. Derivation of the Adjustable Robust Counterpart 278
D. Derivation of the Static Robust Counterpart 281
E. DCVRP Formulation 284

Bibliography 286
LIST OF FIGURES

1.1 Power demand and power generation from solar and wind energy in California on April 15, 2015. Data are provided by the California Independent System Operator (CAISO, 2015). .................................................. 3
1.2 Two perspectives on DSM, which requires both the grid operator and the consumer to interact through physical grid operations and the electricity markets. ................................................................. 5
1.3 Classification of DSM activities. Rectangular boxes depict DSM programs introduced by the grid operator, rounded rectangles indicate measures to be taken by the consumer. .......................... 7
2.1 Data points are sampled from the feasible region. The nonconvex feasible region can be approximated more accurately by the union of multiple convex regions. ................................................................. 31
2.2 (a): Points marked by circles (center area) have the same linear cost correlation. The remaining points, marked by diamonds, have a different linear cost correlation. (b) and (c): Assignment of data points and resulting polytopes from Phase 1 and 2. .................................................. 35
2.3 Flowchart for the Phase 1 algorithm .................................................. 36
2.4 For the illustrative example, the constraint set given by (2.6) with $m = 3$ has many solutions which lead to overlapping convex hulls. 37
2.5 The added cuts eliminate assignment solutions guaranteed to result in overlapping convex hulls based on information from the current solution. ................................................................. 40
2.6 Starting from the convex hull of the given data points, the Phase 2 algorithm first finds the contour of the tight envelope around the feasible region and subsequently constructs the polytopes representing the envelope. ................................................................. 41
2.7 Overview of the Phase 2 algorithm. .................................................. 42
2.8 Flowchart of the algorithm applied to each facet $f \in \bar{F}$ to find new vertices and create new facets. ................................. 43
2.9 All data points lie on one side of a supporting hyperplane. .... 45
2.10 The dashed line indicates the old facet which led to the two new facets. 45
2.11 A candidate point is found if the polytope formed with the facet does not contain any data points in its interior. ......................... 46
2.12 F is the new vertex originating from facet A-B. E is not a candidate point for facet B-C because the resulting polytope B-C-E and the already cut-off polytope A-B-F overlap. ................................. 48
2.13 The data points marked by diamonds are the candidate points for the facet at the top. From all candidate points, the one marked by a hollow diamond is the one furthest away from the facet and is therefore declared as a new vertex. ................................. 49
2.14 New facets created at each iteration. The thick red lines show the newly added facets whereas the dashed red lines indicate the facets in the previous iteration from which the new ones originate. .... 50
2.15 Flowchart of the algorithm applied to construct the desired convex regions. ......................................................... 51
2.16 Contour of the feasible region with labeled vertices. .......... 54
2.17 A region is convex if and only if the hyperplanes containing the assigned facets are supporting hyperplanes for the region. .... 54
2.18 The convex region assignment problem may yield overlapping convex regions. By adding the proposed cuts, these solutions can be avoided. ................................................................. 55
2.19 In this example, point B is a redundant vertex because it has been declared a vertex, but is not a vertex of any of the constructed convex regions. ......................................................... 56
2.20 Subset assignments resulting from Phase 1 for the 2D cases with 221 data points. ......................................................... 60
2.21 Subset assignments resulting from Phase 1 for the 3D cases with 214 data points. ......................................................... 61
2.22 Convex regions resulting from Phase 2 for the 2D cases with 91 data points. ......................................................... 62
2.23 Convex regions resulting from Phase 2 for the 2D cases with 167 data points. ......................................................... 63
2.24 Convex regions resulting from Phase 2 for the 3D cases with 114 data points .......................................................... 65
2.25 Convex regions resulting from Phase 2 for the 3D cases with 234 data points .......................................................... 66
2.26 In Phase 1, the given data points are partitioned into two disjoint subsets, denoted subsets 1 and 2. ......................... 67
2.27 Depending on $\delta$, the Phase 2 algorithm constructs different convex regions. ..................................................... 69

3.1 Common-grid time representation with a time period length of $\Delta t$ and the present time point defined as time point 0. .......... 75
3.2 Simple process network consisting of process nodes (rectangles), material nodes (circles), and arcs depicting the material flows. .... 75
3.3 This illustrative example shows the feasible operating region of a process that can operate in three different operating modes. ....... 77
3.4 Mode transition graph showing the different operating modes and all possible transitions with the corresponding operational constraints. 79
3.5 In this illustrative example, the cumulative electricity purchase meter is read every four hours. ............................... 81
3.6 With a discount contract, unit price decreases with increasing electricity purchase. The ranges for the different prices define the contract blocks. ..................................................... 82
3.7 A simple penalty contract can be modeled as a three-block contract, with the first and third blocks corresponding to under- and overconsumption, respectively. ........................................ 83
3.8 The process network of the illustrative example consists of 4 process and 7 material nodes. ........................................ 86
3.9 Product demand profiles for the three different scenarios (low, high, and medium). .................................................. 86
3.10 Gantt chart for the optimal schedule in Case 1. Selected operating modes are shown for each time period. ....................... 90
3.11 Amount of electricity consumed by each process in Case 1. ........ 90
3.12 Inventory profiles of the intermediate materials D and F, and the final products E and G in Case 1. ............................. 91
3.13 Amount of electricity consumed by each process in Case 2. ........ 91
3.14 Amount of electricity consumed by each process in Case 3. ........ 92
3.15 Breakdown of the total electricity purchase into the purchases from
the TOU and the discount contracts in Case 4. .......................... 92
3.16 Breakdown of the total electricity purchase into the purchases from
the TOU and the penalty contracts in Case 5. .......................... 92
3.17 Process network representing the given air separation plant. ....... 93
3.18 Amount of electricity consumed by each process of the air separation
plant. ...................................................................................... 94
3.19 Electricity prices for the TOU contract, the penalty contract, and the
spot market. .............................................................................. 95
3.20 Breakdown of the total electricity purchase into the purchases from
the three difference sources. ...................................................... 95
3.21 GN2 production and its breakdown into feeds for different processes. 96
3.22 LN2 production and inventory profile. ...................................... 96

4.1 Contract prices typically consist of a time-dependent and an
amount-dependent component. .................................................. 105
4.2 CVaR can be defined in terms of cost or in terms of profit, which can
have different implications on the solution. ............................... 108
4.3 Flowchart for Benders decomposition algorithm. ......................... 114
4.4 Spot electricity price profiles associated with the scenario set with
medium level of price uncertainty. ............................................... 116
4.5 Electricity purchase profiles for the medium-Var case. ............... 119
4.6 Electricity purchase profiles for the medium-Var medium-Var case. 123
4.7 Spot electricity price profiles associated with the full set of scenarios
and the expected price profile. .................................................... 124
4.8 Spot electricity price profiles associated with the reduced scenario set. 124
4.9 Electricity purchase profiles for the high-Var case. ...................... 126
4.10 Solutions obtained by changing the weights in the objective function,
showing the trade-off between total expected profit and CVaR. ....... 128

5.1 Integrated ASU-CES system. The mass and power flows are de-
picted by solid and dashed lines, respectively. Flow variable names
are shown in parentheses. .......................................................... 139
5.2 For the illustrative example, the diagram shows a possible realiza-
tion of the uncertainty if the uncertainty set given by Eq. (5.13) with
\( \Gamma = 3 \) is applied. ............................................................... 146
5.3 For the illustrative example, the diagram shows a possible realization of the uncertainty if the uncertainty set given by Eq. (5.14) with \( \Gamma = [1, 1, 1, 2, 2, 2, 3, 3] \) is applied. .................. 146

5.4 For the benchmark case only considering the ASU, the optimal solution suggests load shifting toward low-price time periods and shutting down the plant for 54 hours in the middle of the week. ...... 150

5.5 In the benchmark case with integrated ASU and CES, the plant operates longer and part of the electricity consumed is recovered from the CES. ................................. 151

5.6 The change in product inventory levels shown in the diagrams is due to the flows into and out of the inventory tanks. ................. 152

5.7 Liquid flow into the CES tank increases the CES inventory while it depletes when stored liquid is converted to power either to be used internally or sold to the market. ....................... 152

5.8 In Case 1, by participating in the reserve market, more power is recovered from CES for internal use. ........................................ 153

5.9 For Case 1, the CES inventory profile is shown for the scenario in which no reserve power is dispatched. Reserve capacity is provided during time periods in which the spinning reserve price is high. ... 154

5.10 CES inventory levels for Case 2 (top diagram) and Case 3 (bottom diagram). ......................................................... 155

5.11 In general, relative cost savings increase with increasing efficiency and decreasing utilization. ........................................ 155

6.1 The V-representation makes use of the polytope’s vertices; the H-representation is formed by the supporting hyperplanes. ...... 167

6.2 Electricity and interruptible load prices for the illustrative example. .. 173

6.3 Product flows and resulting inventory profile for the case without interruptible load. .................................................. 173

6.4 Target electricity consumption profile and provided interruptible load for the case of \( \bar{\zeta} = 0 \), and price profiles. ......................... 175

6.5 Target and recourse product flows and target inventory profile for the case of \( \zeta = 0 \). ................................................. 175

6.6 Target and recourse product flows and target inventory profile for the case of \( \bar{\zeta} = 47 \). ............................................. 176
6.7 Target and recourse product flows and target inventory profile for the case of \( \zeta = 47 \) and \( \Gamma = [1, 2, \ldots, 48]^T \). ........................................... 178
6.8 Absolute cost savings vs. level of plant utilization in the illustrative example. ..................................................... 179
6.9 Product flows and inventory profiles for the case without interruptible load. ....................................................... 180
6.10 Target electricity consumption profile and provided interruptible load for the case of \( \zeta = 23 \), and price profiles. ............... 181
6.11 Target and recourse product flows and target inventory profile for the case of \( \zeta = 23 \). .......................................................... 182
6.12 Absolute cost savings vs. level of plant utilization in the industrial case study. ...................................................... 182

7.1 Fine and coarse time grids for a planning horizon of 12 h with \( \Delta t^f = 1 \) h and \( \Delta t^c = 4 \) h. .................................................. 191
7.2 Network representation of the distribution model. Each node corresponds to a plant \( p \), route \( s \), or customer \( c \). The notation of the flow variables is shown on the top arcs. .................................................. 195
7.3 Supply chain network for the illustrative example, with 2 plants and 50 customers. ...................................................... 204
7.4 Feasible operating regions of the production modes given for the two plants in the illustrative example. ........................... 205
7.5 Production quantities, shipments, and inventory levels of products I1 and I2 at Plant P1 in the illustrative example. ........... 206
7.6 Production quantities, shipments, and inventory levels of products I1 and I2 at Plant P2 in the illustrative example. ........... 207
7.7 Optimal routing solution for the illustrative example. ........... 207
7.8 Evolution of the routing solution obtained from applying the proposed heuristic algorithm to the illustrative example. .... 208
7.9 Electricity consumption and electricity price profiles for each plant. .... 217
7.10 Production quantities, shipments, and inventory levels of LO2 and LN2 at Plant P1. .................................................. 218
7.11 Production quantities, shipments, and inventory levels of LO2 and LN2 at Plant P2. .................................................. 219
7.12 Comparison of the numbers of customers to be visited on each day of the planning horizon as suggested by Heuristics PH1, PH2, and H3. 220
7.13 Comparison of the numbers of plant-to-customer allocation changes from the current assignment required for Heuristics PH1, PH2, and H3. ................................................................. 221

8.1 Timeline showing some of the seminal works in flexibility analysis (FA), robust optimization (RO), and robust optimization in PSE. . . . . . 230
8.2 Examples of feasible and infeasible design with respect to a given uncertainty set \( T \). ................................................................. 232
8.3 Illustrative example with \( n_\theta = 1 \), where \( \chi(d) = \bar{\chi}(d) \). ................. 238
8.4 Illustrative example with \( n_\theta = 2 \), where \( \chi(d) < \bar{\chi}(d) \). ................. 239
8.5 Example 1, HEN with four uncertain inlet temperatures and one control variable (\( Q_C \)). ................................................................. 248
8.6 Example 2, simple process with uncertainty in product demands and equipment geometries. ...................................................... 251
8.7 Schematic of the petrochemical complex considered in Example 3 (figure adapted from Park et al. (2006)). For the detailed process network, see the original paper by Sahinidis et al. (1989). . . . . . . . . . 252

A.1 CRS expressed as the difference of the convex hull around all data points and the union of infeasible (empty) convex regions. . . . . . . . . 275
LIST OF TABLES

1.1 Overview of reviewed works. The papers are listed in chronological order, and the various model features are shown in thematic groups.  13
1.2 Overview of thesis chapters 2–8, listing which ones of the four main challenges are addressed in each chapter. ......................... 25

2.1 Computational results from Phase 1 for the 2D cases .................. 60
2.2 Computational results from Phase 1 for the 3D cases .................. 61
2.3 Computational results from Phase 2 for the 2D cases. .................. 64
2.4 Computational results from Phase 2 for the 3D cases. .................. 65
2.5 Computational results from Phase 2 for the 4D cases .................. 66
2.6 Constants and coefficients for the power consumption correlations. 68

3.1 Vertices associated with each operating subregion of the processes from the illustrative example. ................................. 87
3.2 Electricity consumption correlations associated with each operating subregion. Each correlation is a linear function of the materials. . . . 88
3.3 Possible transitions between the different operating modes of each process and the corresponding minimum stay times. .................. 88
3.4 Predefined sequences of mode transitions and the corresponding fixed stay times. ......................................................... 89
3.5 Inventory bounds and initial inventory levels for each material. ...... 89
3.6 The table lists the five cases, which differ in the product demands and in the power contracts, with the optimal total electricity costs. .. 89

4.1 Vertices associated with each operating mode of the plant from the illustrative example. .................................................. 115
4.2 Coefficients for linear electricity consumption correlations associated with each operating mode. ........................................ 115
4.3 Inventory bounds and initial inventory levels for each product. ...... 116
<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>Demand values and probabilities for each scenario of the three different demand scenario sets.</td>
<td>117</td>
</tr>
<tr>
<td>4.5</td>
<td>Expected costs and VSS resulting from risk-neutral optimization with only electricity price uncertainty.</td>
<td>118</td>
</tr>
<tr>
<td>4.6</td>
<td>Expected costs and VSS from risk-neutral optimization with only demand uncertainty.</td>
<td>119</td>
</tr>
<tr>
<td>4.7</td>
<td>Expected costs and VSS from risk-neutral optimization with both electricity price and demand uncertainty.</td>
<td>120</td>
</tr>
<tr>
<td>4.8</td>
<td>Expected profits, CVaRs, and VSS from risk-averse optimization with only electricity price uncertainty.</td>
<td>121</td>
</tr>
<tr>
<td>4.9</td>
<td>Expected profits, CVaRs, and VSS from risk-averse optimization with only demand uncertainty.</td>
<td>121</td>
</tr>
<tr>
<td>4.10</td>
<td>Expected profits, CVaRs, and VSS from risk-averse optimization with both electricity price and demand uncertainty.</td>
<td>122</td>
</tr>
<tr>
<td>4.11</td>
<td>Expected costs and VSS from risk-neutral optimization.</td>
<td>125</td>
</tr>
<tr>
<td>4.12</td>
<td>Expected profits, CVaRs, and VSS from risk-averse optimization.</td>
<td>126</td>
</tr>
<tr>
<td>4.13</td>
<td>VSS in % obtained from risk-averse optimization for different demand distributions and levels of demand uncertainty.</td>
<td>127</td>
</tr>
<tr>
<td>4.14</td>
<td>Computational results for risk-neutral optimization.</td>
<td>129</td>
</tr>
<tr>
<td>4.15</td>
<td>Computational results for risk-averse optimization with ζ = 0.5 and α = 0.9.</td>
<td>130</td>
</tr>
<tr>
<td>6.1</td>
<td>H-representations, fixed (δₘ) and unit (γₘ) electricity consumption for each operating mode (product and subregion indices have been omitted).</td>
<td>172</td>
</tr>
<tr>
<td>6.2</td>
<td>Possible transitions between the different operating modes and the corresponding minimum stay times.</td>
<td>172</td>
</tr>
<tr>
<td>6.3</td>
<td>Costs and revenues in $ for cases with different δ. Here, C^{PD}, C^{PC}, R^{IL}, C^{RC}, and TC denote production cost, purchasing cost, revenue from providing interruptible load, recourse cost, and total cost, respectively.</td>
<td>177</td>
</tr>
<tr>
<td>6.4</td>
<td>Total costs, model sizes, and computation times for instances with different δ.</td>
<td>181</td>
</tr>
<tr>
<td>7.1</td>
<td>Comparison of costs and number of candidate routes in the full MPRP formulation and the restricted MPRPs solved in the heuristic algorithm.</td>
<td>209</td>
</tr>
</tbody>
</table>
LIST OF TABLES

7.2 Overview of generated MPRP instances, grouped into five data sets. . 210
7.3 Parameter settings for Heuristic H2. . . . . . . . . . . . . . . . . . . . . . 212
7.4 Parameter settings for Heuristic H3, with $k$ being the iteration counter. 212
7.5 Comparison of solutions for all instances in Set A. . . . . . . . . . . . . . . . 213
7.6 Comparison of solutions for all instances in Set B. . . . . . . . . . . . . . . . 213
7.7 Comparison of solutions for all instances in Set C. . . . . . . . . . . . . . . . 214
7.8 Comparison of solutions for all instances in Set D. . . . . . . . . . . . . . . . 214
7.9 Comparison of solutions for all instances in Set E. The average values for
Heuristic H2 are computed over the available numbers. . . . . . . . . . . . . 214
7.10 Comparison of costs and solution times for the industrial test case. . . . . 219
8.1 Flexibility test results for Example 1. . . . . . . . . . . . . . . . . . . . . . 249
8.2 Flexibility index results for Example 1. . . . . . . . . . . . . . . . . . . . . . 249
8.3 Data for discrete scenarios considered in Example 1. . . . . . . . . . . . . . 249
8.4 Design under uncertainty results for Example 1. . . . . . . . . . . . . . . . 250
8.5 Flexibility test and flexibility index results for Example 2. . . . . . . . . . 252
8.6 Flexibility test results for Example 3. . . . . . . . . . . . . . . . . . . . . . 255
8.7 Flexibility index results for Example 3. . . . . . . . . . . . . . . . . . . . . . 256
8.8 Design under uncertainty results for Example 3. . . . . . . . . . . . . . . . 257
B.1 Complete set of data for the illustrative example. . . . . . . . . . . . . . . . 277
B.2 Cost constants and coefficients used in the illustrative example. . . . . . 277
C.1 List of lower-level problems . . . . . . . . . . . . . . . . . . . . . . . . . . . 279
C.2 Dual formulations of lower-level problems . . . . . . . . . . . . . . . . . . . . 280
1. Introduction

Demand side management (DSM), which refers to the active management of electricity demand, has been recognized as an effective approach to improving power grid performance and consumer benefits. For electricity consumers, DSM constitutes the opportunity to benefit from financial incentives by adjusting their electricity consumption. Especially for the chemical industry, which is a major electricity consumer, DSM is becoming increasingly critical for maintaining profitability.

In this work, we take a systems approach to industrial DSM, focusing on decision-making at the enterprise level. The objective is to develop systematic optimization tools for solving DSM problems that arise when operating industrial power-intensive processes. Due to the time-sensitive pricing and availability of electricity, we have to consider operational decisions over an extended time horizon. Here, the accurate representation of process flexibility and dynamics is crucial. Furthermore, production scheduling has to be integrated with energy management, which requires the modeling of various electricity market mechanisms. Another major focus of this work is the treatment of uncertainty. In industrial DSM, we encounter various sources of uncertainty, with the most notable being electricity price. In order to consider uncertainty in the decision-making process, we explore different modeling techniques, namely stochastic programming and robust optimization. Finally, following the enterprise-wide optimization paradigm, in addition to planning and scheduling at the plant level, the goal is to also consider DSM at the supply chain level, which introduces logistics as an additional element in the integrated optimization framework.

In this introductory chapter, we first describe in Section 1.1 the challenges in the modern power grid that call for improved operations in order to ensure grid reliability and sustainability. Section 1.2 introduces the concept of the so-called smart grid, which involves DSM as one of its essential components. We propose a definition of DSM and a classification of DSM activities in Section 1.3 before highlighting the distinguishing characteristics of industrial DSM in Section 1.4.
subsequent comprehensive review of existing systematic approaches to planning and scheduling for industrial DSM, presented in Section 1.5, we identify four major challenges in this area. Addressing theses challenges is the goal of this thesis, for which an outline is presented in Section 1.6.

1.1. Challenges in the Modern Power Grid

The power grid is designed to reliably match electricity supply and demand. This task has become increasingly challenging due to high fluctuations in electricity demand and increasing penetration of intermittent renewable energy into the electricity supply mix.

According to the latest annual energy outlook of the U.S. Energy Information Administration (EIA, 2015), about 40% of the total current U.S. energy consumption is attributed to electricity generation. In the EIA’s business-as-usual projection, the total electricity consumption in the U.S. is expected to increase from 3904 GWh in 2015 to 4797 GWh in 2040. Due to the steady growth in energy demand and the pressure to reduce greenhouse gas emissions, many efforts are being made toward using more renewable energy sources for electricity generation.

The largest growth is seen in electricity generation from solar and wind energy. These renewable energy sources are ecologically sustainable and can potentially constitute a major share of the energy mix; however, they are also intermittent in nature, i.e. the energy output varies, often irregularly as it depends on weather conditions. The intermittency of renewable electric power generation is a huge challenge for power grid operations as the availability of these energy sources does not match the electricity demand. Figure 1.1 shows the power demand as well as the power generation from solar and wind in California on April 15, 2015. Following a typical trend, the power demand is low at night, increases in the morning, remains high during the day until it reaches its peak in the evening, and decreases afterwards. Naturally, solar power generation only occurs during the day and is zero at night. In particular, one can see that no solar energy is available during peak-demand hours. Wind power generation is usually highest at night, which also applies in this particular case.

Net load is defined as the difference between electricity demand and electricity production from variable noncontrollable generation resources; it represents the demand portion that has to be met by conventional controllable power plants. The discrepancy between power demand and renewable power generation results in a
1. INTRODUCTION

Figure 1.1: Power demand and power generation from solar and wind energy in California on April 15, 2015. Data are provided by the California Independent System Operator (CAISO, 2015).

net load curve that reaches its low in the early afternoon and peaks in the evening. The peak load determines the required total power generation capacity. The minimum net load is typically met by efficient but very inflexible base load power plants, such as nuclear and coal-fired plants. The difference between the minimum and maximum net loads is the demand that has to be met by so-called peaking power plants, which can quickly vary their power output but have higher unit costs than base load power plants. Hence, flattening the net load curve is a major goal in power systems design and operation. A flatter net load curve with lower peak demand reduces the need of building new power plants and the required share of peaking power plants.

Another fundamental challenge is uncertainty. Apart from the power output of intermittent renewable energy sources, power demand is also extremely difficult to predict. Furthermore, due to its large size and complexity, the power grid is susceptible to transmission congestion and equipment failure, which can lead to brownouts and blackouts. To ensure grid reliability in the presence of this high level of uncertainty, large reserve capacities are required. These reserve capacities are resources that can quickly react to contingency events and eliminate large electricity supply-demand gaps.

The deregulation of electricity markets in the U.S. started in the 1990s. Since then, the experience has shown that allowing more participants, and hence more competition in the electricity markets, can significantly increase grid efficiency. However, deregulation also creates further uncertainty, primarily reflected in the high variability in real-time electricity prices.
1.2. Smart Grid and Demand Side Management (DSM)

In recent years, the notion of a smart grid has been evolving, which represents the concept of a power grid in which the major operations—electricity generation, transmission, distribution, and consumption—are executed in a coordinated and efficient manner. To establish such a smart grid, three essential components are required: (1) an information and communications infrastructure for the collection and real-time exchange of data on grid conditions, (2) advanced decision-making tools for the optimization of grid operations using the collected information, and (3) efficient electricity markets providing appropriate financial incentives that encourage the various participants, e.g. generators and consumers, to actually implement the necessary operations.

The idea of a smart grid has gained considerable interest in industry, research, and public policy. Since 2009, the U.S. Department of Energy and the electric power industry have jointly invested over $7.9 billion in smart grid projects as part of the Smart Grid Investment Grant Program (DOE, 2013), and similar efforts have been undertaken in many other countries.

Traditionally, in power systems engineering, the focus has been on improving the power supply for given electricity demands (loads) in the grid. A major innovation in smart grid is to also include the management of flexible loads, which is generally referred to as demand side management (DSM) since it involves the enhancement of energy systems on the electricity demand side. DSM is expected to play a crucial role in the improvement of grid efficiency and reliability as well as the creation of additional benefits for the consumers (Levy, 2006; Strbac, 2008; Siano, 2014); this has spurred tremendous research efforts across multiple disciplines, such as electrical engineering, civil and environmental engineering, economics, data science, behavior science, and engineering public policy.

The opportunities in DSM can be viewed from two distinct perspectives: the grid operator’s perspective and the electricity consumer’s perspective. On the one hand, the grid operator’s main objective is to increase efficiency and ensure stability in the power grid. In this context, DSM is regarded as a means to reduce the overall electricity demand, to flatten the load curve and hence reduce the required peak generation capacity, as well as to provide the flexibility to quickly react to supply-demand mismatch in the grid by adjusting loads. On the other hand, the electricity consumer’s objective is simply cost reduction. For electricity consumers, DSM is required in order to adapt to price signals coming from the electricity mar-
1. Introduction

Illustration 1.2: Two perspectives on DSM, which requires both the grid operator and the consumer to interact through physical grid operations and the electricity markets.

The electricity consumer’s perspective has to be considered in order to achieve a more accurate assessment of individual consumers’ DSM potentials. This knowledge will consequently lead to more active participation of consumers and make
electricity markets more efficient and competitive (Kirschen, 2003). Here, domain knowledge is required since each process has its own operational limitations, costs, safety requirements, etc. Also, different preferences in terms of convenience and risk have to be taken into account.

One distinguishes between three consumer sectors: residential, commercial, and industrial. DSM in the first two sectors deals with residential and commercial buildings (Motegi et al., 2007), in which load adjustment is mainly achieved by controlling the HVAC and lighting systems, whereas industrial DSM is concerned with power-intensive industrial processes (Samad & Kiliccote, 2012).

1.3. Definition and Classification of DSM Activities

When it comes to defining DSM, most descriptions represent the grid operator’s perspective. For example, in a report released by The World Bank (Charles River Associates, 2005), DSM is defined as the

systematic utility and government activities designed to change the amount and/or timing of the customer’s use of electricity for the collective benefit of the society, the utility and its customers.

Notice that according to this definition, only activities on the utility and government sides are considered DSM, whereas electricity consumers take a rather passive role and only react to those DSM activities. Although such a definition is perfectly correct and underlines the origin of DSM as a concept proposed by utilities (Gellings, 1985), it understates the degree of freedom that consumers have in their decision-making. In fact, only the consumers can actually change their electricity consumption; utilities and governments can only provide incentives that encourage such activities. Hence, a more comprehensive definition of DSM considering both perspectives could be:

DSM encompasses systematic activities at the interplay between grid operator and electricity consumer aiming at changing the amount and/or timing of the consumer’s use of electricity in order to increase grid performance and consumer benefits. DSM activities on the grid operator side involve the assessment of the need for load adjustment and the creation of financial incentives for the consumer, while the consumer reacts to these financial incentives and performs the actual physical load adjustment operations.
Note that depending on the level of regulation in the electricity market, the grid operator could be an independent organization or the electric utility itself. Depending on the size, an electricity consumer could be one individual consumer or an aggregator that manages many small consumers.

Figure 1.3 shows a general classification of DSM activities, which consist of various DSM programs introduced by the grid operator (rectangular boxes) and the measures that need to be taken by the consumer (rounded rectangles) in order to participate in these DSM programs. The two main DSM categories are energy efficiency (EE) and demand response (DR) (Charles River Associates, 2005). The goal of EE is to reduce energy consumption while accomplishing the same tasks, and DR refers to load profile adjustment, such as load shifting and load shedding, driven by market incentives.

**Figure 1.3:** Classification of DSM activities. Rectangular boxes depict DSM programs introduced by the grid operator, rounded rectangles indicate measures to be taken by the consumer.

In DR, one distinguishes between dispatchable and nondispatchable DR (FERC, 2010), which are often also referred to as incentive-based and price-based DR (DOE, 2006), respectively. Dispatchable DR refers to load adjustment capacities that consumers provide to the grid operator such that these capacities can be dispatched to maintain grid stability or in times of emergency. The grid operator has control over dispatchable DR resources by either direct load control or by requesting the con-
sumers to reduce their power consumption (interruptible load) when a DR event, e.g. a generator failure, occurs. The various types of dispatchable DR resources, many of which classified as ancillary services, mainly differ in the amount of time within which the consumer has to respond to DR requests. In general, the faster one can react, the more valuable is the DR service, i.e. the more consumers are rewarded for providing such DR capacities.

Nondispatchable DR resources are not controlled by the grid operator; instead, consumers choose to adjust their power consumption profiles based on price signals from the electricity market. Time-of-use (TOU), critical peak, and real-time pricing are just three of many pricing schemes designed to encourage consumers to change their load profiles according to the power grid’s needs.

EE can be primarily achieved by improved process design or retrofit of the existing process that results in higher efficiency. Also, efficiency can be increased by optimal scheduling and control strategies that maximize the time in which the process runs at its most energy-efficient operating point. Effective planning, scheduling, and control are even more critical in DR. Here, operational flexibility is key. In nondispatchable DR, electricity is treated as any other commodity that can be purchased, but with two distinct characteristics: it is difficult to store electricity, and electricity prices are extremely volatile. Hence, processes have to be flexible in order to react to price changes. In dispatchable DR, consumers are rewarded not so much for the actual dispatch of DR resources but for the capability of quickly reacting to DR events whenever they occur. Providing dispatchable DR requires a very high degree of flexibility in the consumer’s process since requests by the grid operator, which typically cannot be anticipated in advance, have to be met while maintaining process feasibility and safety.

1.4. Characteristics of Industrial DSM

Although there are large untapped DSM potentials in all three—residential, commercial, and industrial—sectors (Gellings et al., 2006), there are some distinguishing features of industrial processes that facilitate the deployment of DSM strategies:

- In the industrial sector, individual power consumption is very high, which motivates and eases participation in DSM programs. For example, aluminum production has an energy intensity of 71 GJ per tonne (Worrell et al., 2008), and a typical aluminum plant produces hundreds of tonnes of aluminum on a daily basis.
• In most cases, advanced metering infrastructure is already in place; therefore, the capital investment required to implement DSM in industry is close to zero.

• Industrial processes operate in isolated environments such that human comfort is usually not an issue; this is in contrast to the residential and commercial sectors, where e.g. HVAC control is constrained by the maximum decrease in comfort induced by temperature changes.

However, industrial processes are often highly complex and subject to strict safety requirements. In industrial DSM, it is therefore crucial to carefully evaluate the flexibility of each process in order to avoid detrimental disruptions caused by sudden changes in the plant operation.

In the following, we list some distinct characteristics of industrial electricity consumers, which need to be considered in the DSM decision-making process:

• Many manufacturing processes are highly integrated and have critical temporal dependencies, which have to be taken into account when operating the plant; this requires deep process knowledge.

• While direct load control is common in the residential sector, it usually cannot be applied in industry because of safety considerations.

• Electricity is difficult to store; however, most commodity products are not. Product inventory naturally increases the flexibility in plant operations and therefore allows more room for DSM.

• Industrial electricity consumers often enter into power contracts that offer special rates under given conditions.

• Large industrial power-intensive plants often have substantial onsite electricity generation capacities. Some of the generated electricity may even be sold at the market price or transferred offsite.

• Industrial usage data are typically confidential since they could reveal competition-sensitive information on operations strategies and process performance. Hence, all DSM efforts have to be managed within the same company.
According to the EIA (2012), the total net electricity demand by the U.S. industry in 2010 amounted to 850 TWh, with the five most power-intensive sectors—chemicals, primary metals, paper, food, and petroleum and coal products—combined consuming 560 TWh. Highly power-intensive processes include gas compression, electrolysis, and electric heating. Samad & Kiliccote (2012) present five real-world case studies in which industrial DSM has been successfully implemented and has generated considerable cost savings. Among the examples, the most prominent case is the one of Alcoa (Todd et al., 2009), which uses the operational flexibility of its aluminum smelting facilities to provide ancillary services through which the company is achieving large economic benefits.

1.5. Advances and Challenges in Planning and Scheduling for Industrial DSM

Because of the strong dependence of electricity price, electricity availability, and DR events on time, effective planning and scheduling tools are essential in DSM, especially in an industrial setting where complex manufacturing processes are involved. Production scheduling has been an active field of research in operations research since the 1950s (Graves, 1981), and it started to attract increased attention in the process systems engineering (PSE) community in the 1970s (Reklaitis, 1982). Since then, considerable progress has been made in the modeling of production scheduling problems as well as in the development of efficient methods for solving these models. For recent reviews of works on production scheduling in PSE, we refer to Méndez et al. (2006), Maravelias (2012), and Harjunkoski et al. (2014). Furthermore, Maravelias & Sung (2009) discuss the integration of scheduling and planning, which involves longer time horizons; and Li & Ierapetritou (2008a) and Verderame et al. (2010) review approaches proposed for scheduling under uncertainty.

For industrial DSM, we can leverage the tremendous advances in production planning and scheduling made over the last few decades; however, there are unique challenges that require special attention. The four main challenges that we see in planning and scheduling for industrial DSM are the following:

1. **Modeling operational flexibility**

   Electricity prices are extremely time-sensitive. In a typical day-ahead market, the price varies from hour to hour; in the real-time market, it changes every few minutes. In order to capture this time dependence and determine
the ability of a process to quickly respond to price changes, very detailed scheduling models are required. In these models, the representation of time and the accurate modeling of constraints on transitions between different process operating points are especially critical.

2. *Integration of production and energy management*

Traditionally, production and energy management are handled separately. For planning and scheduling, this means that first, a production scheduling problem is solved, and once the production schedule is determined, the objective of energy management is to minimize the cost for purchasing the electricity required for this particular production schedule. This sequential approach can easily lead to suboptimal solutions since possible synergies between production and energy management are not taken into account. Hence, an integrated approach that considers both parts simultaneously can be very beneficial, especially when power contracts with complex constraints are applied. Moreover, energy management can be further complicated by the presence of onsite generation and participation in dispatchable DR programs.

3. *Decision-making across multiple time scales*

To perform DSM scheduling, a detailed model with a fine time representation is required; the typical time horizon for such a scheduling problem is one day or one week. In contrast, in long-term planning, the time horizon may span multiple months or years. In that case, however, we cannot simply apply the same detailed model with an extended time horizon because the resulting model would be computationally intractable, nor can we use an aggregate model with a coarse time representation since then we would not be able to model DSM activities. Hence, computationally efficient planning models have to be developed that can capture both long-term as well as short-term effects.

4. *Optimization under uncertainty*

Due to the high level of uncertainty in the power grid, electricity prices are extremely volatile. Also, all dispatchable DR activities are intrinsically uncertain because the consumer does not know in advance when the grid operator will request the dispatch of those DR services. Furthermore, uncertainties on the production side, e.g. regarding product demand and processing time, also exist. The major challenge lies in the accurate characterization of the relevant sources of uncertainty and optimal decision-making while considering
1. INTRODUCTION

these uncertainties.

Associated with all the above is the challenge of computational efficiency. With the incorporation of new features, the models become more complex and computational tractability becomes an issue. Many large-scale real-world problems cannot be solved by using off-the-shelf tools. Therefore, along with novel modeling approaches, efficient solution methods have to be developed in order to improve the computational performance.

In the following, we present a comprehensive review of existing works addressing the aforementioned four main challenges. All reviewed works are listed in Table 1.1, presented in chronological order with respect to the publication date. The table shows various features of the different models, and we will refer back to this overview in the next subsections.

1.5.1. Modeling Operational Flexibility

The key to industrial DSM is operational flexibility, which allows load adjustment in response to electricity market signals. In this context, a production facility’s operational flexibility is mainly defined by its ability to ramp up and down production and its product inventory capacity. In order to assess the potential benefits from DSM for an industrial plant, a detailed scheduling model capturing all relevant process constraints and interactions with electricity markets is required. The development of such scheduling models has been identified as a promising research topic only very recently. From Table 1.1, one can see that almost all works addressing this subject have been conducted after year 2000, with the vast majority published within the past five years.

Relevant Industrial Processes

Industrial processes considered in the literature can be grouped into two general categories: continuous production and batch production. For example, air separation and aluminum manufacturing are typical continuous processes, whereas steel production is mainly operated in batch mode. Table 1.1 shows for each reference whether the proposed model is primarily designed for continuous or batch production processes. Note that some of the models can also be applied to model hybrid (continuous and batch) production environments.

Also, Table 1.1 lists particular industrial processes to which each model has been applied. Cryogenic air separation and steel manufacturing, which are ar-
Table 1.1: Overview of reviewed works. The papers are listed in chronological order, and the various model features are shown in thematic groups.

<table>
<thead>
<tr>
<th>Model Features</th>
<th>Continuous Production</th>
<th>Batch Production</th>
<th>Air Separation</th>
<th>Alumina</th>
<th>Chlorine</th>
<th>Kraft</th>
<th>Bleaching</th>
<th>Power Contracts</th>
<th>Stochastic Optimization</th>
<th>Discrete-Time</th>
<th>LP</th>
<th>NLP</th>
<th>MILP</th>
<th>MINLP</th>
<th>CP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daryanian et al. (1989)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ashok &amp; Banerjee (2001)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ierapetritou et al. (2002)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Everett &amp; Philpott (2002)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ashok (2006)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Karwan &amp; Keblis (2007)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Babu &amp; Ashok (2008)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Castro et al. (2009)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Yusta et al. (2010)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Nolde &amp; Morari (2010)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hait &amp; Artigues (2011)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Castro et al. (2011)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fang et al. (2011)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mitra et al. (2012a)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mitra et al. (2012b)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Wang et al. (2012)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Vujanic et al. (2012)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Castro et al. (2013)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Artigues et al. (2013)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Tan et al. (2013)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ding et al. (2014)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Wang et al. (2014)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Mitra et al. (2014)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hadera et al. (2014)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Zhang &amp; Hug (2014)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Shrouf et al. (2014)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Zhang &amp; Hug (2015)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hadera et al. (2015)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hadera et al. (2015)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Tan et al. (2015)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Guarably the most complex production processes among the ones listed, have been considered more extensively. Cryogenic air separation is power-intensive because
of the large amount of compression that is required in the separation and liquefaction processes. In steel manufacturing, the most power-intensive production stages are the melting process in the electric arc furnace and the hot rolling process.

Other power-intensive industrial processes considered in case studies are aluminum, cement, chlor-alkali, flour, and pulp production, and machining. Aluminum and chlor-alkali manufacturing involve electrolysis, while the high power intensities in cement, flour, pulp, and machining processes stem from mechanical operations such as grinding, milling, and turning.

Prevalent Modeling Concepts

In one of the first works on DSM scheduling, Daryanian et al. (1989) propose a multiperiod model that merely consists of inventory constraints and bounds on the production rate. However, industrial processes are seldom that simple, and more accurate representations require detailed models involving more complex constraints.

In cases where the scheduling problem is primarily concerned with the sequencing and timing of power-intensive production tasks, typical machine scheduling formulations have been applied in several proposed models (Nolde & Morari, 2010; Fang et al., 2011; Wang et al., 2012). Tan et al. (2013) and Hadera et al. (2015) further include constraints on waiting times between consecutive production stages, which are especially important in steel manufacturing.

Many chemical production environments exhibit a network structure (Maravelias, 2012) in which material handling constraints play an essential role. The scheduling of such processes is the focus of the works by Ashok and coworkers (Ashok & Banerjee, 2001; Ashok, 2006; Babu & Ashok, 2008), who emphasize the impact of storage in industrial DSM. For the same purpose, Ding et al. (2014) apply the well-known concept of the state-task network (STN) (Kondili et al., 1993; Shah et al., 1993), in which state nodes represent feeds, intermediates, and final products, and task nodes represent processing operations. A similar concept is the one of the resource-task network (RTN) (Pantelides, 1994), which forms the basis for several DSM scheduling models proposed by Catro and coworkers (Castro et al., 2009, 2011, 2013).

Another popular modeling concept is based on the notion of operating modes. First proposed by Ierapetritou et al. (2002) and further developed by Karwan & Keblis (2007), it takes into account that production and power consumption characteristics can vary within the same process depending on the configuration or state
in which the process is operating. In a mode-based model, the process can only operate in one of the given operating modes, which represent particular operating states such as “off”, “on”, or “startup”. Each mode is defined by a specific feasible region in the product space and a power consumption function with respect to the production rates. Mitra et al. (2012a, 2013) reformulate the model by Karwan & Keblis (2007) to improve the tightness of the formulation, and develop additional constraints to impose restrictions on the transitions between different modes. A mode-based model has also been applied by Shrouf et al. (2014).

**Time Representation**

One important attribute of scheduling models is the representation of time, which is especially critical in DSM applications because of the highly time-sensitive nature of electricity prices. In general, one distinguishes between discrete- and continuous-time models, and there is a large body of work in the literature discussing different formulations and their strengths and limitations (Floudas & Lin, 2004; Méndez et al., 2006; Sundaramoorthy & Maravelias, 2011).

As shown in Table 1.1, predominantly discrete-time models have been used in DSM scheduling. In a discrete time representation, in which the scheduling horizon is divided into time periods of known lengths, it is straightforward to model the time-varying electricity price by simply assigning different price values to different time periods. In most cases, hourly electricity prices are considered, with a scheduling horizon of one day or one week resulting in 24 or 168 time periods, respectively.

Unlike discrete time representation, continuous time representation allows processing tasks to start at any point in the continuous time domain. In scheduling problems in which tasks can change within small time intervals, continuous-time models can be beneficial since an appropriate discrete-time model may require a very fine time discretization, which could dramatically increase the size of the model. However, with time-varying electricity prices, modeling the electricity cost becomes a challenge in continuous-time formulations.

Castro et al. (2009) propose a continuous-time formulation in which varying electricity prices are defined over price periods with different starting and ending times, and tasks are disaggregated into tasks executed at different electricity prices. Nolde & Morari (2010) propose a continuous-time model for load tracking in which the electricity consumption in prespecified load intervals is determined by computing the overlap of tasks with the load intervals. The same concept can
be applied to determine electricity cost when electricity prices vary with time (Tan et al., 2013). Nolde & Morari (2010) present a formulation using binary variables and corresponding big-M constraints for the different task-interval overlap cases. This formulation has been improved by Haït & Artiges (2011) who compute the overlaps by introducing binaries indicating whether a task begins before or during a price interval. This reformulation greatly reduces the number of constraints and binary variables. A continuous-time model incorporating a similar approach has been proposed by Hadera et al. (2015). While applying the same concept to maintenance scheduling of a gas engine power plant, Castro et al. (2014) propose a further improved reformulation derived by using generalized disjunctive programming techniques (Grossmann & Trespalacios, 2013) and by introducing redundant constraints.

Computational performance is often the key criterion when choosing between discrete- and continuous-time scheduling models. An analysis of the reviewed works shows that at this point, discrete-time models generally show better computational performance when applied to large-scale problems. Castro et al. (2009) show that only problems of small size can be handled effectively by the continuous-time model, while problems of industrial significance can be solved efficiently with a comparable discrete-time model. In order to mitigate this limitation, Castro et al. (2011) propose an aggregate discrete-time model that is used in conjunction with a continuous-time model in a rolling horizon framework. The computational results show that the proposed solution approach is considerably more efficient than both traditional full-space discrete- and continuous-time models. However, under restricted power availability, the rolling horizon approach may lead to suboptimal solutions, in which case the full-space discrete-time model becomes the better choice. Hadera et al. (2015) apply a heuristic bilevel decomposition approach to solve the proposed continuous-time model. The decomposition approach significantly reduces the solution time; however, the problem is still only tractable for a single-day scheduling horizon.

Types of Models

From Table 1.1, we can see that the vast majority of the reviewed models are formulated as mixed-integer linear programs (MILPs). Two very simplistic models (Daryanian et al., 1989; Wang et al., 2014) are formulated as linear programs (LPs). A nonlinear programming (NLP) formulation is proposed by Yusta et al. (2010) for machining process scheduling, where the nonlinearity stems from the equation ex-
pressing the lifetime of the cutting tool as a nonlinear function of the cutting speed and the power consumption function.

Ierapetritou et al. (2002) apply a quadratic power consumption function, which gives rise to a mixed-integer nonlinear programming (MINLP) formulation. Generalized Benders decomposition (Geoffrion, 1972) and outer approximation (Duran & Grossmann, 1986) have been applied to solve the problem. Since the MINLP is convex in the continuous variables, both solution algorithms are guaranteed to obtain the global optimal solution; however, they require considerable computational expense for solving industrial-scale problems. Hence, Ierapetritou et al. (2002) create an approximate MILP by linearizing the power consumption function and solve the MILP instead of the original MINLP. The results show that the MILP model can be solved in significantly less computation time and obtains solutions that are very close to the true optimal solutions of the MINLP. Babu & Ashok (2008) propose an MINLP model with quadratic functions expressing the power factors and efficiencies of each subprocess. The model has been solved with the global optimization algorithm implemented in the LINDO solver.

Moreover, Artigues et al. (2013) show an interesting application of constraint programming (CP) in industrial DSM. Here, the scheduling problem is solved in two steps. In the first step, a job assignment and sequencing problem with fixed job durations is solved with a CP model; then in the second step, an MILP scheduling model is solved with the job assignment and sequencing obtained in the first step. Obviously, this is another attempt to reduce the computational effort by decomposing the problem.

Main Insights from Case Studies

A number of case studies have been presented in the reviewed references, some using real-world data from industry. Here, we summarize some of the main insights drawn from these case studies in which scheduling under time-sensitive electricity prices has been considered.

All case study results show the high potential benefit of industrial DSM, accomplished primarily by shifting load toward low-price periods. Cost savings up to 20% can be achieved compared with scheduling assuming constant electricity prices (Castro et al., 2009). The optimization even shows that in certain cases, it can be beneficial to shut down the plant for a long period of time (Ierapetritou et al., 2002; Mitra et al., 2012a). However, the impact of DSM strongly depends on the level of operational flexibility. In particular, if a plant is highly utilized, i.e. it has
to operate at close to full capacity in order to meet the product demand, there will be hardly any room for load shifting (Ashok, 2006; Mitra et al., 2012a). Therefore, the benefit of nondispatchable DR usually decreases with increasing level of plant utilization; however, this may be different in the case of dispatchable DR.

1.5.2. Integration of Production and Energy Management

Industrial DSM relies on the integrated optimization of production and energy management. In most existing works on DSM planning and scheduling, the energy management part only consists of purchasing electricity at time-varying prices with possibly an upper bound constraint on the amount of electricity that can be purchased in each time period. As indicated by the overview shown in Table 1.1, only very recently, more complex energy management activities involving power contracts, onsite generation, and dispatchable DR have been considered.

Nolde & Morari (2010) consider a load tracking problem in which the actual electricity consumption of a steel plant is supposed to match a committed load curve; penalties incur for over- and underconsumption. Besides load tracking, the models proposed by Hadera et al. (2014, 2015) account for multiple electricity sources as well as onsite generation, which generates electricity that can be either used to power the steel plant or sold to the electricity market. A network flow formulation is applied to incorporate the electricity purchase and sales options into the scheduling model. Furthermore, Hadera et al. (2015) have developed a mean value cross decomposition approach that can help reduce the computational effort for solving such large-scale integrated problems. Results from case studies show that often a substantial amount of electricity is generated onsite, some of which is sold to the market in order to reduce the net electricity cost. Also, the penalties for deviating from the committed load curve can be quite significant and are often in the same order of magnitude as the net electricity cost. Onsite generation is also considered in the model proposed by Wang et al. (2012), which further includes fuel storage constraints and gas emissions in the objective function.

Vujanic et al. (2012) and Zhang & Hug (2014, 2015) develop systematic approaches for the optimization of dispatchable DR activities, e.g. the provision of regulation and operating reserve services. Participation in dispatchable DR programs creates new unconventional revenue streams that can significantly reduce net operating costs; however, the process is associated with high degree of uncertainty due to the unpredictability of DR events.
1.5.3. Decision-making Across Multiple Time Scales

In long-term DSM planning, tactical and strategic decisions may be considered, such as investment decisions for capacity expansion and retrofit, and the selection of long-term power contracts. Since planning problems involve much longer time horizons, they are typically solved using models that are considerably less detailed than short-term scheduling models. However, such aggregate models cannot be applied to industrial DSM problems because they do not capture time-sensitive electricity prices and DR events with sufficient accuracy.

Integrated decision-making across multiple time scales—short-term operational and long-term tactical and strategic DSM decisions—has barely been considered in the literature. Mitra et al. (2014) propose a multiscale capacity planning model for power-intensive continuous processes considering hourly changes in electricity price. The objective is to find the optimal investment strategy for purchasing new equipment, performing equipment upgrades, and installing additional storage facilities over a planning horizon of multiple years. Instead of applying a detailed representation across the entire time horizon, which would be computationally intractable, the model is simplified by leveraging the seasonality of electricity prices. Here, each year is divided into four seasons, and each season is represented by one week, which is repeated cyclically and characterized by a typical electricity price profile that reflects the price’s seasonal behavior. An hourly time discretization is applied, which results in 672 time periods representing each year (4 seasons, each with a week divided into 168 time periods). While the number of time periods is rather large, it is considerably smaller compared with the 8760 time periods that would be required to represent hourly discretization over one year.

1.5.4. Optimization Under Uncertainty

Most existing planning and scheduling tools for DSM are deterministic, i.e. they assume that all given input parameters are certain, including future electricity prices. However, this assumption is rarely valid, especially in the case of spot electricity prices, which are very difficult to forecast (Zareipour et al., 2010). In the light of the high level of uncertainty in DSM problems, optimization models have been developed that consider the characteristics of the uncertain parameters instead of simply assigning to them their expected values. In this way, a more realistic representation of the problem can be achieved, which forms the basis for improved
decision-making.

Two major uncertainty modeling approaches have been applied to DSM planning and scheduling problems: stochastic programming (Birge & Louveaux, 2011), and robust optimization (Ben-Tal et al., 2009). In stochastic programming, the uncertainty is represented by discrete scenarios with given probabilities, and decisions are made at different stages, which are defined such that realization of uncertainty is observed between two stages. At each stage, actions depending on previous observations are taken; such reactive actions are also referred to as recourse decisions. In robust optimization, the uncertainty is specified in terms of an uncertainty set from which any point is a possible realization of the uncertainty. A robust optimization model is formulated such that it is feasible for the entire uncertainty set and optimizes the worst case.

**Uncertainty in Electricity Price and Product Demand**

Ierapetritou et al. (2002) present a two-stage stochastic programming framework in which the electricity prices for the first three days of the scheduling horizon are assumed to be known while the prices for the remaining days are assumed to be stochastic. The uncertain prices are characterized by a set of scenarios with each scenario corresponding to a particular price profile for the time beyond the first three days. While the first-stage decisions are related to the first three days, the second-stage decisions are related to the remaining days of the scheduling horizon and can be different for each scenario. Given probabilities for all scenarios, the objective is to minimize the total expected operating cost. A similar approach is taken by Everett & Philpott (2002) who assume that all future electricity prices are uncertain; hence, each scenario represents a price profile over the entire scheduling horizon.

Monte Carlo simulation with a stochastic price forecasting model can be used to generate electricity price profiles that are needed in a scenario-based approach. However, in order to accurately characterize the price uncertainty over a longer period of time, many price profiles are required, which leads to a large-scale stochastic programming model that may be computationally intractable. This limitation has motivated Mitra et al. (2012b) to apply a robust optimization approach to model uncertain electricity prices, which features different possible price ranges and accounts for correlated data.

Since DSM comprises both production and energy management, only accounting for uncertainty related to electricity is often insufficient. Uncertainty on the
production side may have a different and possibly larger impact on the plant operations, and decision-making in the presence of multiple sources of uncertainty is certainly nontrivial. In their proposed stochastic programming model, Mitra et al. (2014) consider product demand uncertainty, which is a parameter that can have a profound impact on the solution and is often associated with high degree of uncertainty. For different cases, Mitra et al. (2014) compute the value of stochastic solution (VSS), which measures the improvement in the objective function value achieved by solving the stochastic model compared with the solution obtained from the deterministic model using mean values. The results show that the VSS can be quite significant, especially for skewed demand distributions with large standard deviations.

**Uncertainty in Dispatchable DR**

The development of systematic decision-making tools for dispatchable DR has not been attempted until recently. The main challenge lies in the inherent uncertain nature of the problem since the consumer does not know in advance when the dispatch of the provided DR resources will be requested.

Zhang & Hug (2014) apply a stochastic programming approach to optimize the provision of regulation capacity by aluminum smelters. Likewise, by applying a scenario-based approach similar to the one proposed by Conejo et al. (2002) for electricity producers, Zhang & Hug (2015) derive a bidding strategy for aluminum smelters. In the bidding process, participants state how much energy or operating reserve capacity they are willing to sell at which price. In the proposed stochastic programming framework, the price is the uncertain parameter, and a scheduling problem is solved for each price scenario. The solution provides price-amount pairs for each scenario, which can be used to construct the bidding curve.

In the above-mentioned stochastic programming approaches, the same probabilities are assumed for all scenarios. This assumption is usually not realistic; in fact, it is very difficult to obtain reasonable probability distributions for dispatchable DR events. Furthermore, when providing operating reserve, dispatch upon request has to be guaranteed since otherwise, one has to pay very high penalties. Hence, robust solutions are required. Vujanic et al. (2012) consider uncertainty in the start times of scheduled tasks, which may be caused by load shifting required to meet operating reserve demand. Robust optimization has been applied to ensure feasibility for any changes in task start times within prespecified ranges.
1.6. Outline of the Thesis

The objective of this work is to take a systems approach to industrial DSM and develop systematic decision-making tools that address the four major challenges presented in the previous section. As we take the perspective of the chemical industry and make decisions at the enterprise level, the concepts applied in this work fall under the general framework of enterprise-wide optimization (EWO) (Grossmann, 2005). In this context, we consider various aspects concerning process modeling, planning and scheduling, supply chain management, and optimization under uncertainty. Although most of the proposed models and algorithms can be extended to solve more general problems, here we focus on their application to DSM-specific problems. In the following, we provide an overview of the remaining chapters of this thesis.

In Chapter 2, the challenge of process modeling in an EWO context is considered. Many chemical processes are highly complex and exhibit nonlinear behavior, which usually requires computationally intensive models to be described accurately. However, one cannot afford embedding such detailed process models in higher-level scheduling and supply chain models for obvious computational reasons. Hence, there is a need for developing surrogate process models that are sufficiently accurate as well as computationally efficient. In Chapter 2, we propose an algorithm for the data-driven construction of a type of surrogate model that can be formulated as a set of mixed-integer linear constraints, yet still provide good approximations of nonlinearities and nonconvexities. In such a surrogate model, which we refer to as Convex Region Surrogate (CRS), the feasible region is given by the union of convex regions in the form of polytopes, and for each region, the corresponding cost function can be approximated by a linear function. The general problem is as follows: Given a set of data points in the parameter space and a scalar cost value associated with each data point, find a CRS model that approximates the feasible region and cost function indicated by the given data points. To the best of our knowledge, the problem as such has not been reported before in the literature. We present a two-phase algorithm to solve this problem and demonstrate its effectiveness in an extensive computational study as well as a real-world case study.

In Chapter 3, a general discrete-time model is proposed for the scheduling of power-intensive process networks with various power contracts. The proposed model consists of a network of processes represented by CRS models that are in-
corporated in a mode-based scheduling formulation. Furthermore, a block contract model is considered that allows the modeling of a large variety of commonly used power contracts. The resulting MILP model is applied to an illustrative example as well as a real-world industrial test case. The results demonstrate the model’s capability of representing the operational flexibility in a process network and different electricity pricing structures.

In Chapter 4, we consider the problem of risk-based integrated production scheduling and electricity procurement. The proposed two-stage stochastic programming model accounts for the two most critical sources of uncertainty that occur in this context: spot electricity price and product demand. Conditional value-at-risk is incorporated into the model as a measure of risk. Furthermore, scenario reduction and multicut Benders decomposition are implemented to solve large-scale real-world problems. The proposed model is applied to an illustrative example as well as an industrial air separation case. In the analysis of the results, we emphasize the difference between risk-neutral and risk-averse optimization.

A very different source of uncertainty is considered in Chapter 5. Here, we take a look at the interesting concept of storing energy in the form of liquefied gas, referred to as cryogenic energy storage (CES). We realize that although CES on an industrial scale is a relatively new approach, the technology is well-known and essentially part of any air separation unit that utilizes cryogenic separation. We assess the operational benefits of adding CES to an existing air separation plant, investigating three new potential opportunities: (1) increasing the plant’s flexibility for load shifting, (2) storing purchased energy and selling it back to the market during higher-price periods, and (3) creating additional revenue by providing operating reserve capacity. Here the uncertainty lies in the reserve demand. Since demand for operating reserve occurs due to unexpected events, the reserve provider does not know in advance when and how much reserve service has to be dispatched; however, reserve dispatch has to be guaranteed upon request. We apply a robust optimization approach that enforces feasibility for all possible reserve demand scenarios defined by a given uncertainty set.

Operating reserve can be provided not only by power generation facilities, but also by consumers that can quickly ramp down their electricity consumption. The latter type of operating reserve is also called interruptible load. In Chapter 6, we develop an adjustable robust scheduling model for continuous industrial processes providing interruptible load. Unlike in Chapter 5, where we apply static robust optimization, which does not account for recourse, the proposed adjustable robust
model incorporates recourse decisions by expressing them as linear functions of the uncertain parameters. We emphasize that being able to take uncertainty-dependent decisions is crucial in the context of interruptible load. In the case studies, we show the change in the quality of the solution with varying extent of recourse.

In Chapter 7, we consider industrial DSM at the supply chain level. This problem is motivated by the complexity of industrial gas supply chains, which consist of multiple power-intensive air separation plants, possibly each with a very different power contract, and hundreds of customers to which products are delivered using tanker trucks. The goal is to simultaneously make detailed scheduling decisions on the production side and tactical planning decisions on the distribution side while considering vendor-managed inventory; hence, we need to solve a multiscale production routing problem. The major challenge lies in the integrated optimization of production and distribution operations, where the latter can only be accurately modeled by incorporating routing decisions. For large-scale instances, the resulting models cannot be solved to optimality by using existing exact solution methods; hence, we propose an iterative MILP-based heuristic that can obtain high-quality solutions in reasonable computation times. The effectiveness of the proposed solution method is demonstrated in an extensive computational study. Furthermore, we show that when applied to a real-world industrial test case, our approach outperforms state-of-the-art solution methods currently considered in practice.

Similar to robust optimization, flexibility analysis is an approach to solving optimization problems under uncertainty. In fact, flexibility analysis and robust optimization share some fundamental concepts, such as the use of polyhedral uncertainty sets and the worst-case approach to guarantee feasibility. In Chapter 8, we examine the connection between these two approaches, which has not been sufficiently acknowledged in the literature. First, we compare flexibility analysis and robust optimization from a historical perspective. Then, for linear systems, new formulations for the three classical flexibility analysis problems—flexibility test, flexibility index, and design under uncertainty—based on duality theory and the affinely adjustable robust optimization approach are proposed. We show that the latter approach is generally more restrictive such that it may lead to overly conservative solutions. The proposed formulations are applied to three numerical examples, and the results show improved computational performance compared with the traditional flexibility analysis models.

Table 1.2 provides an overview of Chapters 2–8, specifying which ones of the four aforementioned challenges in industrial DSM are the focus of each chapter. Fi-
nally, in Chapter 9, we provide a critical review of the thesis, along with a summary of the main contributions and suggestions for future work.

**Table 1.2:** Overview of thesis chapters 2–8, listing which ones of the four main challenges are addressed in each chapter.

<table>
<thead>
<tr>
<th>#</th>
<th>Chapter Title</th>
<th>Process Modeling</th>
<th>Integrated Optimization</th>
<th>Multiscale Optimization</th>
<th>Uncertainty</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Data-driven Construction of Convex Region Surrogate Models</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Scheduling of Continuous Power-intensive Process Networks with Various Power Contracts</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Risk-based Integrated Production Scheduling and Electricity Procurement</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Robust Scheduling of Air Separation Plants with Cryogenic Energy Storage</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Adjustable Robust Scheduling of Continuous Industrial Processes Providing Interruptible Load</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Multiscale Production Routing in Supply Chains with Power-intensive Production Facilities</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>On the Relation Between Flexibility Analysis and Robust Optimization for Linear Systems</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note that a vital part of this work is the analysis of the results from real-world case studies for which the data are provided by Praxair; however, due to confidentiality reasons, we cannot disclose detailed information about plant specifications and actual product demands. Therefore, all results from the industrial case studies are given as dimensionless quantities, and numerical values are normalized if necessary.
2. **Data-driven Construction of Convex Region Surrogate Models**

In order to consider integrated optimization problems involving complex processes, we need process models that are sufficiently accurate as well as computationally efficient. In this work, we develop an algorithm for the data-driven construction of a type of surrogate model that can be formulated as a set of mixed-integer linear constraints, yet still provide good approximations of nonlinearities and nonconvexities. In such a surrogate model, which we refer to as Convex Region Surrogate (CRS), the feasible region is given by the union of convex regions in the form of polytopes, and for each region, the corresponding cost function can be approximated by a linear function. The general problem is as follows: Given a set of data points in the parameter space and a scalar cost value associated with each data point, find a CRS model that approximates the feasible region and cost function indicated by the given data points. We present a two-phase algorithm to solve this problem and demonstrate its effectiveness in an extensive computational study as well as a real-world case study.

This chapter is organized as follows. Section 2.1 provides a review of related works and further motivates the problem. In Section 2.2, we explain the concept of CRS models and their advantages when embedded in MILP formulations. In Section 2.3, we formally state the problem of constructing CRS models. Section 2.4 shows a brief illustrative overview of the proposed algorithm that is divided into two phases, which are described in detail in Sections 2.5 and 2.6. After summarizing the complete algorithm in Section 2.7, we present the results from an extensive computational study and apply the proposed method to a real-world industrial case study in Section 2.8. In Section 2.9, we comment on some limiting features of the proposed algorithm, before closing the chapter with a summary of the main contributions and results in Section 2.10.
2.1. Background

This chapter deals with the following problem: Given a set of data points in the $K$-dimensional parameter space and scalar cost values associated with each data point, find a set of convex regions in the $K$-dimensional space such that the union of the convex regions describes a tight envelope around all data points, and for the data in each region, find a linear cost correlation within a prespecified error tolerance. This is a fairly generic problem and in our proposed algorithm, we apply concepts from various areas, such as computational geometry, mixed-integer programming, polyhedral theory, metamodeling (Simpson et al., 2001), clustering (Jain, 2010), and pattern recognition (Jain et al., 2000). However, to the best of our knowledge, the problem as such has not been reported before in the literature.

This work is motivated by the high demand for computationally efficient process models in many engineering applications. Typically, when constructing a mathematical model, we have to trade off model accuracy against computational tractability, especially if we want to use the model for optimization purposes. If the computational budget forbids the use of a detailed process model, we have to create a surrogate model that can be used instead. As the trend goes toward integrated optimization involving multiple levels of decision-making, e.g. in dynamic real-time optimization or planning and scheduling, there is an increasing demand for high-quality surrogate models that are computationally efficient and at the same time ensure a sufficient degree of accuracy (Chung et al., 2011).

Surrogate modeling techniques can be divided into two general categories: model order reduction and data-driven modeling (Biegler et al., 2014). In model order reduction, one starts with a high-fidelity model and tries to reduce the complexity of that model while retaining most of the structure of the original equations. Data-driven modeling, on the other hand, solely relies on data and does not make use of the explicit formulation of an existing model. Data-driven modeling approaches have become popular in recent years because of their wide applicability (Queipo et al., 2005; Wang & Shan, 2007; Cozad et al., 2014) and are used in one of the following three situations: (1) The existing model is too complex to be reduced to the desired degree by using model order reduction techniques. (2) There exists a “black box” model that can be used for simulation, but the model formulation is not given in explicit form. (3) A mathematical model of the process does not exist, yet we can draw data from the actual process.

A major challenge in surrogate modeling is the approximation of nonlinearities
and nonconvexities, in which case most existing methods have to apply nonlinear structures in order to achieve good accuracy. However, in mixed-integer programming, there is typically a very significant difference in computational complexity between a linear and a nonlinear formulation. Thus, we want to avoid nonlinearities when constructing surrogate models for mixed-integer programming frameworks, which frequently arise in various engineering applications, such as process design, planning and scheduling, and supply chain optimization.

Ierapetritou (2001) applies the concept of flexibility analysis (Swaney & Grossmann, 1985a) and uses the Quickhull algorithm (Barber et al., 1996) to find a convex hull as approximation of the feasible region of a given model. The constructed convex hull is guaranteed to be inscribed in the actual feasible region, however, only if the feasible region is convex. To evaluate the feasibility of nonconvex processes, Goyal & Ierapetritou (2003) develop an approach in which the feasible region is approximated by subtracting outer polytopes around the infeasible regions from an expanded convex hull obtained by simplicial approximation (Goyal & Ierapetritou, 2002). Sung & Maravelias (2007) propose an attainable region approach that determines the feasible production levels of the underlying state-task production scheduling model and an underestimation of the production cost in an integrated planning and scheduling framework. In a subsequent work (Sung & Maravelias, 2009), the same authors develop an extension of the attainable region approach that takes nonconvexities into account by combining multiple polytopes. It should be noted that these methods become computationally expensive in cases of higher dimensions and with nonconvexities.

The above approaches all require the full mathematical formulation of the model that we want to approximate. Despite the general applicability of data-driven methods, the number of related works taking an approach similar to ours is very limited. There is a large body of work on data-driven techniques in the machine learning (Mitchell, 1997; Hastie et al., 2009) community. However, most loosely related existing machine learning methods, such as support vector machines and k-means clustering, are designed for classification or pattern recognition purposes and cannot be directly applied to solve the given problem. Several works (Üney & Türkay, 2006; Xu & Papageorgiou, 2009; Kone & Karwan, 2011) have proposed mixed-integer programming approaches to multi-class data classification in which hyperboxes are used to define the regions for the different classes. Karwan & Keblis (2007) use the convex hull around given data points as an approximation of the feasible region and apply linear regression to find a linear surrogate
cost correlation. This method is easy to implement and has been successfully applied in other works such as Mitra et al. (2012a); however, the resulting surrogate model is only accurate if the true feasible region is convex and the true cost function is linear.

In this work, we propose to approximate the feasible region with the union of convex regions in the form of polytopes. Furthermore, for each region, the corresponding cost function is approximated by a linear function. We show that such a surrogate model, which we refer to as Convex Region Surrogate (CRS), can be formulated as a set of mixed-integer linear constraints. Thus, CRS models are ideal for problems that are already formulated as MILPs since in that case, including a CRS model in the formulation of the optimization problem will not increase its structural complexity.

The main contribution of this work is the development of an algorithm for the data-driven construction of CRS models. The proposed algorithm takes data drawn from the process (or simulations), and constructs convex regions such that the union of the convex regions describes a tight envelope around all data points and a sufficiently accurate linear approximation of the cost function can be found for each region. As will be shown, this is a challenging problem that may require significant computational effort.

### 2.2. Convex Region Surrogate

To motivate the construction of CRS models, let us consider the following mixed-integer program:

\[
\begin{align*}
\min & \quad \alpha^T u + \beta^T w + \sum_{i \in I} f_i(x_i) \\
\text{s.t.} & \quad A u + D w + \sum_{i \in I} E_i x_i \leq b \\
& \quad u \in \mathbb{R}^m, w \in \mathbb{Z}^n \\
& \quad x_i \in X_i \quad \forall i \in I
\end{align*}
\]

(2.1)

where \( u \) are continuous variables and \( w \) are integer variables. Each variable vector \( x_i \) is bounded by a set \( X_i \), and \( f_i(x_i) \) is the corresponding cost function. Both \( X_i \) and \( f_i \) can be nonlinear and nonconvex. We have fixed constraint matrices and vectors \( A, D, E_i, \) and \( b \), and fixed cost coefficients \( \alpha \) and \( \beta \).

From the structure of the problem in (2.1), we can see that the major difficulty
in terms of computational complexity will arise from the nonlinearities in $X_i$ and $f_i$, especially if the set $I$ is large. A typical example is a discrete-time scheduling formulation, in which $I$ is the set of time periods. The variable vector $x_i$ could represent the amounts of products produced in time period $i$, and $f_i(x_i)$ could be the corresponding cost. The set $X_i$, which bounds $x_i$, contains all constraints related to the production process. Eq. (2.1b) would represent constraints linking the different time periods. Since the structures of $X_i$ and $f_i$ have a large impact on the computational tractability of the problem, it is desirable to replace them by computationally more efficient approximations without losing much accuracy in the model.

The concept of the CRS is inspired by Karwan & Keblis (2007) who propose to approximate the feasible region $X$ with the convex hull around all given data points, and apply linear regression to the data to find a linear approximation of the cost correlation $f$. The advantage of this approach is that the model remains linear and convex. Also, one can easily reduce the dimension of the feasible region by only considering relevant variables, which is especially useful in multiscale optimization. For example, in production scheduling, we often only need to know how much the plant can produce. Thus, instead of considering all variables including temperatures, pressures, and intermediate flows, we would only include the production rates in the surrogate model.

The approach by Karwan & Keblis (2007) has two major limitations. First, if the true cost correlation is nonlinear, linear regression over all given data points will be inaccurate. Second, if the true feasible region is nonconvex, the convex hull around all data points will provide a poor approximation. As an example, suppose we want to approximate the shaded nonconvex feasible region shown in Figure 2.1a, and we are given the data points shown in the same figure. If we just take the convex hull, we obtain the approximation shown in Figure 2.1b, which is not very accurate.

Instead of using the convex hull, we propose to approximate the feasible region with the union of convex regions in the form of polytopes. In this way, as illustrated in Figure 2.1c, we can obtain a considerably more accurate representation of the feasible region. In addition, if the true cost function is not linear, we can construct the regions such that a sufficiently accurate linear approximation of the cost correlation can be found for the data of each region. In this way, we obtain a piecewise linear approximation of the cost function.

Since each region is a polytope, any point in a region can be expressed as a
2. Data-driven Construction of CRS Models

Figure 2.1: Data points are sampled from the feasible region. The nonconvex feasible region can be approximated more accurately by the union of multiple convex regions.

A convex combination of its vertices. A feasible point has to be in one of the regions. Furthermore, the form of the cost function depends on the region in which the feasible point lies. As a result, the CRS model can be naturally described with the following disjunction:

\[
\begin{bmatrix}
Y_r \\
x = \sum_{j \in V_r} \lambda_j v_{rj} \\
\sum_{j \in V_r} \lambda_j = 1 \\
0 \leq \lambda_j \leq 1 & \forall j \in V_r \\
f = b_r + c_r^T x
\end{bmatrix}
\]

\[
\bigvee_{r \in R} Y_r
\]

\[Y_r \in \{\text{true, false}\} \quad \forall r \in R\]

where \(R\) is the set of convex regions and \(V_r\) is the set of vertices of region \(r\). In each region \(r\), \(x\) is described as the convex combination of vertices \(v_{rj}\) with \(j \in V_r\). The nonnegative multipliers, which have to sum up to 1 over \(j \in V_r\), are denoted by \(\lambda_j\). The cost constant \(b_r\) and coefficients \(c_r\) define the cost correlation in region \(r\). The Boolean variable \(Y_r\) is true if the chosen feasible point lies in region \(r\).

By applying the hull reformulation (Balas, 1985), the disjunction can be transformed into the following set of mixed-integer linear constraints:

\[
f = \sum_{r \in R} (b_r y_r + c_r^T x) \quad (2.3a)
\]
2. Data-driven Construction of CRS Models

\[ x = \sum_{r \in R} \bar{x}_r \]  
(2.3b)

\[ \bar{x}_r = \sum_{j \in V_r} \bar{\lambda}_{rj} v_{rj} \quad \forall \ r \in R \]  
(2.3c)

\[ \sum_{j \in V_r} \bar{\lambda}_{rj} = y_r \quad \forall \ r \in R \]  
(2.3d)

\[ 0 \leq \bar{\lambda}_{rj} \leq 1 \quad \forall \ r \in R, \ j \in V_r \]  
(2.3e)

\[ \sum_{r \in R} y_r = 1 \]  
(2.3f)

\[ y_r \in \{0, 1\} \quad \forall \ r \in R \]  
(2.3g)

where \( x \) and \( \lambda_j \) are disaggregated into the region-dependent variables \( \bar{x}_r \) and \( \bar{\lambda}_{rj} \), respectively. The variables \( \bar{x}_r \) and \( \bar{\lambda}_{rj} \) can only be nonzero if region \( r \) is chosen. The binary variables \( y_r \) have to sum up to 1 over \( r \in R \), i.e. only one region can be chosen. In the cost function, the constants and coefficients also only apply if \( x \) lies in the corresponding region.

By replacing \( X_i \) and \( f_i \) in (2.1) by such CRS models, we arrive at the following formulation:

\[
\begin{align*}
\min & \quad \alpha^T u + \beta^T w + \sum_{i \in I} \sum_{r \in R_i} (b_r y_{ir} + c_r^T x_i) \\
\text{s.t.} & \quad Au + Dw + \sum_{i \in I} E_i x_i \leq b \\
& \quad u \in \mathbb{R}^m, \ w \in \mathbb{Z}^n \\
& \quad x_i = \sum_{r \in R_i} \bar{x}_{ir} \quad \forall \ i \in I \\
& \quad \bar{x}_{ir} = \sum_{j \in V_{ir}} \bar{\lambda}_{irj} v_{irj} \quad \forall \ i \in I, \ r \in R_i \\
& \quad \sum_{j \in V_{ir}} \bar{\lambda}_{irj} = y_{ir} \quad \forall \ i \in I, \ r \in R_i \\
& \quad 0 \leq \bar{\lambda}_{irj} \leq 1 \quad \forall \ i \in I, \ r \in R_i, \ j \in V_{ir} \\
& \quad \sum_{r \in R_i} y_{ir} = 1 \quad \forall \ i \in I \\
& \quad y_{ir} \in \{0, 1\} \quad \forall \ i \in I, \ r \in R_i
\end{align*}
\]  
(2.4a-2.4i)

which is an MILP since the nonlinearities have been replaced by linear constraints.

In summary, the main advantages of CRS models are the following:

- Can approximate nonlinear and discontinuous cost functions.
• Can approximate nonlinear and nonconvex feasible regions.

• By formulating CRS models as sets of mixed-integer linear constraints, efficient MILP solvers such as CPLEX and GUROBI can be used. This is especially useful if the optimization problem, in which we want to integrate the CRS models, is already an MILP.

• The dimension of the variable space in a CRS model can be kept small by only considering data in a subspace, defined by the needs of the particular application or determined by some feature selection mechanism.

**Remark** Instead of expressing the feasible region of a CRS model as the union of feasible convex regions, we can also formulate it as the difference of the convex hull and the union of infeasible convex regions, as shown in Appendix A. This alternative formulation usually gives rise to a weaker MILP formulation; however, it may be the better choice if the number of infeasible convex regions is significantly smaller than the number of feasible convex regions. In the following sections, it will become clear that we can skip the last steps of the algorithm, namely the ones related to the convex region assignment, if we only intend to use the alternative formulation.

In order to formulate a CRS model, we need to determine the set of regions $R$, the corresponding vertices, as well as the region-dependent cost constants and coefficients. Finding these components is a nontrivial task for which we present an algorithm in the remainder of this chapter.

### 2.3. Formal Problem Statement

After we demonstrated in the previous section how to formulate a CRS model as a set of mixed-integer linear constraints given a disjunctive set of convex regions, we take on the task of constructing these convex regions and their cost approximations from data. This problem is formally stated as follows.

Given $n$ data points where each data point $j$ consists of a $K$-dimensional parameter vector $a_j \in \mathbb{R}^K$ and a scalar cost value $g_j$, find convex regions $r$ in the form of polytopes in the parameter space such that

- the union of the convex regions contains all $a_j$;
- no $a_j$ lies in the interior of two or more convex regions, i.e. the convex regions do not overlap except for possibly at the boundaries;
2. DATA-DRIVEN CONSTRUCTION OF CRS MODELS

- in each region \( r \), there exists a linear correlation (with a maximum error \( \epsilon \)) between the parameter and the cost values of the data points contained in region \( r \), i.e.

\[
\bar{g}_j = b_r + c_r^T a_j = b_r + \sum_{k=1}^{K} c_{rk} a_{jk}, \quad |\bar{g}_j - g_j| \leq \epsilon \quad \forall \ j \in J_r
\]  

(2.5)

where \( J_r \) is the set of data points contained in region \( r \);

- the union of the convex regions represents a tight envelope for the feasible region indicated by the given data points.

2.4. Illustrative Overview of the Algorithm

To obtain the CRS, we propose a two-phase algorithm. In Phase 1, the set of all data points is divided into subsets such that a linear parameter-cost correlation can be obtained within a tolerance for all points in each subset, and that the convex hulls constructed around the points of each subset do not overlap. In Phase 2, for each subset, the feasible region indicated by the data points assigned to the subset is approximated by constructing multiple convex regions of which the union contains all points of the subset.

In the next two sections, we present the Phase 1 and Phase 2 algorithms in detail. In order to facilitate the understanding of the algorithm, the explanation of each step is accompanied by an illustrative two-dimensional example. The data points for this example are shown in Figure 2.2a, in which the different markers indicate different parameter-cost correlations. The complete set of data for this illustrative example can be found in Appendix B.

Figures 2.2b and 2.2c show the results at the end of Phase 1 and Phase 2, respectively. In Phase 1, the data points are assigned to three disjoint subsets. Note that we obtain three subsets although there are only two different parameter-cost correlations because with two subsets, the resulting two convex hulls would overlap. In Phase 2, we check for each subset whether it needs to be further partitioned such that multiple convex regions can be obtained to construct a tighter envelope of the feasible region. Here, this is only necessary for the second subset, for which three convex regions are constructed (c.f. Figure 2.2c).
Figure 2.2: (a): Points marked by circles (center area) have the same linear cost correlation. The remaining points, marked by diamonds, have a different linear cost correlation. (b) and (c): Assignment of data points and resulting polytopes from Phase 1 and 2.

2.5. Phase 1: Subset Assignment

In Phase 1, we assign the given data points to subsets such that linear parameter-cost correlations can be obtained for each subset and that the convex hulls of the subsets do not overlap. A flowchart for the Phase 1 algorithm is shown in Figure 2.3. First, we set the initial number of subsets $m$, typically to 1. We then try to assign the data points to $m$ subsets such that linear parameter-cost correlations can be obtained within the tolerance for each subset. We increase $m$ until the assignment problem is feasible. In the next step, the vertices of the convex hulls for all subsets are obtained. Using these, we can then check whether the convex hulls overlap each other. If they do, we add cuts to the subset assignment problem in order to avoid obtaining the same solution in the next iteration. The algorithm terminates when the assignment solution provides $m$ subsets such that the corresponding convex hulls are disjoint.

2.5.1. Subset Assignment Formulation

At the heart of the Phase 1 algorithm is the assignment problem, which assigns $n$ data points to $m$ subsets. For the set of subsets at iteration $t$, $I^t = \{1, 2, \ldots, m\}$, we find a feasible solution to the following set of mixed-integer linear constraints:

$$\sum_{i \in I^t} y_{ij} = 1 \quad \forall \ j$$

(2.6a)
\[
\sum_{j} y_{ij} \geq 1 \quad \forall i \in I^t 
\]  
\[
y_{ij} - \sum_{j', j' < j} y_{i-1, j'} \leq 0 \quad \forall i \in I^t, \; i > 1, \; j 
\]  
\[
\bar{g}_{ij} = b_i + \sum_{k=1}^{K} c_{ik} a_{jk} \quad \forall i \in I^t, \; j 
\]  
\[
\epsilon_{ij} = \bar{g}_{ij} - g_j \quad \forall i \in I^t, \; j 
\]  
\[
-\epsilon - M (1 - y_{ij}) \leq \epsilon_{ij} \leq \epsilon + M (1 - y_{ij}) \quad \forall i \in I^t, \; j 
\]  
\[
y_{ij} \in \{0, 1\} \quad \forall i \in I^t, \; j 
\]

where the binary variable \(y_{ij}\) equals 1 if point \(j\) is assigned to subset \(i\). In this formulation as well as throughout the rest of this chapter, \(M\) denotes a positive big-M parameter. Constraints (2.6a) and (2.6b) ensure that every point is assigned to a subset, and that every subset contains at least one point. Eq. (2.6c) is a symmetry-
breaking constraint that enforces point-to-subset assignment in lexicographic order. Eq. (2.6d) describes the linear cost correlation for each subset $i$ applied to all points $j$, from which the fitting error $\epsilon_{ij}$ is calculated in Eq. (2.6e). The prespecified error tolerance is denoted by $\epsilon$, and constraint (2.6f) forces $\epsilon_{ij}$ to be bounded by $-\epsilon$ and $\epsilon$ if point $j$ is assigned to subset $i$; otherwise, the constraint is relaxed.

Figure 2.4: For the illustrative example, the constraint set given by (2.6) with $m = 3$ has many solutions which lead to overlapping convex hulls.

In general, there are multiple feasible solutions to (2.6), which may lead to the algorithm requiring a large number of iterations if many of the possible solutions do not result in non-overlapping convex hulls. For our illustrative example, Figure 2.4 shows two possible solutions for $m = 3$ in which the resulting convex hulls overlap. Notice that such solutions will be discouraged if we minimize the distances between the points in the same subset. To achieve this clustering effect, we propose the following alternative MILP formulation:

\[
\begin{align*}
\text{min} & \quad \sum_{i \in I^t} \sum_k d_{ik} \\
\text{s.t.} & \quad \text{set of equations (2.6)} \\
& \quad y_{ij} \geq z_{ijj'} \quad \forall \ i \in I^t, \ j, j' > j \\
& \quad y_{ij} \geq z_{ijj'} \quad \forall \ i \in I^t, \ j, j' > j \\
& \quad z_{ijj'} \geq y_{ij} + y_{ij} - 1 \quad \forall \ i \in I^t, \ j, j' > j \\
& \quad -d_{ik} \leq z_{ijj'} (a_{jk} - a_{j'k}) \leq d_{ik} \quad \forall \ i \in I^t, \ j, j' > j, k \\
& \quad y_{ij} \in \{0, 1\} \quad \forall \ i \in I^t, \ j \\
& \quad z_{ijj'} \in [0, 1] \quad \forall \ i \in I^t, \ j, j' > j \\
\end{align*}
\]

where $d_{ik}$ denotes the maximum distance in the $k$th dimension between two points.
in subset $i$, and $z_{ijj'}$ equals 1 if both points $j$ and $j'$ are assigned to subset $i$. Constraints (2.7c)–(2.7e) ensure that $z_{ijj'}$ equals 1 if and only if $y_{ij} = y_{ij'} = 1$. The maximum distances between two points in each subset are determined by including constraint (2.7f) and minimizing $\sum_{i \in I} \sum_{k} d_{ik}$. This formulation facilitates—but does not guarantee—subset assignment that results in non-overlapping convex hulls. In fact, in the case of the illustrative example, it provides the desired solution shown in Figure 2.2b in one iteration. However, (2.7) is a significantly larger problem than (2.6). Therefore, with a large number of data points, it may be computationally more efficient to use (2.6).

2.5.2. Identifying Overlapping Convex Hulls

After constructing the convex hulls, for which we apply the Quickhull algorithm (Barber et al., 1996), we have to check if they overlap. If two convex hulls overlap, we can find a point that belongs to both convex hulls; otherwise, we cannot. We can determine the existence of such a point for each pair of convex hulls $i$ and $i'$ by checking the feasibility of the following set of equations:

\begin{align}
\mathbf{p} &= \sum_{j \in V_i} \lambda_j a_j \quad \text{(2.8a)} \\
\mathbf{p} &= \sum_{j \in V_{i'}} \mu_j a_j \quad \text{(2.8b)} \\
\sum_{j \in V_i} \lambda_j &= \sum_{j \in V_{i'}} \mu_j = 1 \quad \text{(2.8c)} \\
\lambda_j &\geq 0 \quad \forall \ j \in V_i \quad \text{(2.8d)} \\
\mu_j &\geq 0 \quad \forall \ j \in V_{i'} \quad \text{(2.8e)}
\end{align}

where $V_i$ and $V_{i'}$ are the sets of vertices of convex hulls $i$ and $i'$, respectively. Clearly, point $\mathbf{p}$ is constrained to be inside both convex hulls $i$ and $i'$. Thus, (2.8) is only feasible if such a point exists, i.e. if convex hulls $i$ and $i'$ overlap.

Instead of solving (2.8) for each pair of convex hulls, we can also detect all overlapping convex hulls by solving one single LP:

\begin{align}
\min & \quad \sum_{i \in I} \sum_{i' \in I', i < i'} \sum_{k} (s^+_{i'i'k} + s^-_{i'i'k}) \\
\text{s.t. } & \quad p_{i'i'k} = \sum_{j \in V_i} \lambda_{i'i'j} a_{jk} \quad \forall \ i, i' \in I', \ i < i', \ k \quad \text{(2.9a)} \\
& \quad p_{i'i'k} = \sum_{j \in V_{i'}} \mu_{i'i'j} a_{jk} + s^+_{i'i'k} - s^-_{i'i'k} \quad \forall \ i, i' \in I', \ i < i', \ k \quad \text{(2.9b)}
\end{align}
Here we introduce slack variables $s^+_{ii'k}$ and $s^-_{ii'k}$ for each pair of convex hulls $i$ and $i'$ and each dimension $k$. Similar to point $p$ in (2.8), point $p_{ii'}$ is constrained to be inside convex hull $i$. However, according to Eq. (2.9c), point $p_{ii'}$ is only also inside convex hull $i'$ if the corresponding slack variables are zero. LP (2.9) minimizes the sum of all slack variables corresponding to pairs of convex hulls. If a common point can be found, the slacks become zero at the optimal solution and we know that convex hulls $i$ and $i'$ overlap. If they do not overlap, the LP solution yields $\sum_k (s^+_{ii'k} + s^-_{ii'k}) > 0$ since the point $p_{ii'}$ cannot be obtained as a convex combination of vertices of the convex hulls $i$ and $i'$.

The choice of whether to solve (2.8) for $|I|^2!$ pairs of convex hulls or the single LP (2.9) depends on the overhead of solving small LPs in sequence or in parallel and then collecting the results. Often, the overhead time is larger than the benefit we would achieve from solving several small LPs instead of the one large LP.

2.5.3. Overlap Elimination Cuts

The cuts are constraints that are added to the subset assignment problem (2.6) in order to eliminate previous solutions that have led to overlapping convex hulls. In fact, we add cuts that cut off more than just one assignment solution at each iteration. The idea is that if we know from solving (2.9) that convex hulls $i$ and $i'$ overlap, we construct cuts that eliminate all assignment solutions that involve two subsets such that one of the two subsets contains the vertices of convex hull $i$ and the other subset contains the vertices of convex hull $i'$.

We illustrate the proposed cuts with the example in Figure 2.5. Suppose we detected the two overlapping convex hulls, say $i$ and $i'$, in Figure 2.5a. In any assignment solution in which the vertices of convex hull $i$ are assigned to one subset and the vertices of convex hull $i'$ are assigned to another subset, the convex hulls resulting from these subsets will contain convex hulls $i$ and $i'$, and will therefore also overlap each other. This is true regardless which other points are assigned to the subsets. An example is shown in Figure 2.5b where the convex hulls $i$ and $i'$ (dashed lines) are inscribed in the new convex hulls.
2. DATA-DRIVEN CONSTRUCTION OF CRS MODELS

Figure 2.5: The added cuts eliminate assignment solutions guaranteed to result in overlapping convex hulls based on information from the current solution.

The cuts are accumulated in the set $S$, which is set to $\emptyset$ when the number of subsets $m$ is increased, i.e. $S$ is empty at the beginning of each iteration $t$. For each cut $s \in S$, we define a set $I_s = \{i, i'\}$ with $i$ and $i'$ being two convex hulls overlapping each other. The cuts in terms of the assignment variables $y_{ij}$ for a given point $j$ in subset $i$ are formulated as follows:

$$
\sum_{i \in I_s} \sum_{j \in \bar{V}_{is}} y_{ij} \leq \sum_{i \in I_s} |\bar{V}_{is}| - 1 \quad \forall \ s \in S
$$

(2.10)

where $\bar{V}_{is}$ is the set of vertices of convex hull $i$ for cut $s$ and $|\bar{V}_{is}|$ is the cardinality of set $\bar{V}_{is}$. Eq. (2.10) is generally known as a cover cut (Crowder et al., 1983). Note that for each pair of overlapping convex hulls, we have to add $\binom{m}{2}$ cuts in order to account for all possible combinations of subset indices. For example, if we have 3 subsets, we have to add $\binom{3}{2} = 3$ cuts for each pair of overlapping convex hulls, i.e. construct one cut for each of the three subset index pairs $\{1, 2\}, \{1, 3\}$, and $\{2, 3\}$.

2.6. Phase 2: Construction of Convex Regions

In Phase 2, we construct convex regions for each subset $i$ such that the union of the convex regions results in an accurate approximation of the generally nonconvex feasible region. We assume that the feasible region associated with each subset forms a simply connected space, i.e. there are no “holes” in the feasible region.

In our illustrative example, the feasible regions for the first and third subsets are convex such that the convex hulls resulting from Phase 1 are already the tightest approximation (c.f. Figure 2.2b). Therefore, we will demonstrate the Phase 2
2. DATA-DRIVEN CONSTRUCTION OF CRS MODELS

algorithm with the second subset of which the feasible region is clearly nonconvex.

Remark In the following, we allow us a slight misuse of the term “facet”. Strictly speaking, a facet is a feature of a polyhedron; a facet of a polyhedron of dimension $K$ is a $(K - 1)$-dimensional face. Here, we apply the term “facet” in the context of polygons, where it is used as a synonym for “$(K - 1)$-dimensional boundary hypersurface”. The main difference is that for a polyhedron, a facet defines a supporting hyperplane, which does not necessarily hold true in the case of a general polygon.

Phase 2 consists of two main steps as illustrated in Figure 2.6. First, we detect the contour of the feasible region, i.e. we find its vertices and facets. In the second step, the resulting envelope for the feasible region is partitioned into polytopes such that the union of these polytopes forms the desired surrogate feasible region.

![Figure 2.6](image)

**Figure 2.6:** Starting from the convex hull of the given data points, the Phase 2 algorithm first finds the contour of the tight envelope around the feasible region and subsequently constructs the polytopes representing the envelope.

In the following, we will first give an overview of the Phase 2 algorithm and then describe each part in detail. Since Phase 2 can be performed on each subset independently, we omit the index $i$ for the sake of readability.

The high-level flowchart for the Phase 2 algorithm is shown in Figure 2.7. The algorithm is initialized with results obtained in Phase 1, namely the set of vertices $V$ and the set of facets $F$ of the convex hull. The initial $\bar{F}$, which is the set of facets to be examined in the current iteration, is set to $F$. Furthermore, for each facet $f$, we define the set $H_f$ which contains the vertices of facet $f$.

As illustrated in Figure 2.6, the contour of the feasible region can be approximated by facets which are defined by the corresponding vertices. At each iteration, we move into the current outer approximation of the feasible region as far as possible in order to find new vertices and construct new facets using those new vertices. The idea is to take each facet $f \in \bar{F}$ and try to find a data point with which new facets can be formed without cutting off any data points. The point, which fulfills this condition and is sufficiently far away from the facet, is declared a new vertex.
and new facets are formed using this vertex. In this way, a tighter outer approximation of the feasible region is formed. At each iteration, if new facets have been created, the sets $V$, $F$, $\bar{F}$, and $H^f_j$ are updated before the algorithm moves to the next iteration. This process is repeated until no new vertices and facets can be found. Note that after new vertices are added to the initial set $V$, the set of vertices will not define a convex polytope but rather a nonconvex polygon. The algorithm for finding new vertices and creating new facets is described in detail in Section 2.6.1.

After the contour of the feasible region is obtained, the algorithm uses the information about the vertices and facets to construct the desired convex regions. At the heart of this process is an assignment problem, which assigns facets, vertices and data points to convex regions. We elaborate on this part of Phase 2 in Section 2.6.2.

2.6.1. Detecting Contour of the Feasible Region

At each iteration, for each facet $f \in \bar{F}$, the algorithm shown in Figure 2.8 is applied to find one new vertex and create new facets by connecting the new vertex with vertices of facet $f$. We use two criteria to identify the new vertex. First, to obtain an outer approximation of the feasible region, the new facets created using the new vertex must not cut off any data points. Second, to move as far as possible into
the current approximation of the feasible region, we look for the data point that is furthest away from facet $f$ while satisfying the first condition.

![Flowchart](image)

**Figure 2.8:** Flowchart of the algorithm applied to each facet $f \in \bar{F}$ to find new vertices and create new facets.

In the proposed algorithm, we first obtain a unit-length vector that is perpendicular to the facet and points toward the interior of the feasible region. We then find the set of all candidate points $C_f$, where a candidate point is a data point which fulfills the first condition stated above. The normal vector is used in the next step to measure the distance to the facet so that we can determine the point in $C_f$ that is furthest away from the facet. If the distance $d$ is greater than or equal a specified tolerance $\delta$, we declare this point a new vertex, create the corresponding new facets and then move on to the next facet. Otherwise, we directly go to the next facet $f \in \bar{F}$.

**Remark** In practice, we want to avoid creating new facets from facets that are already very small in size. Therefore, in our implementation, we determine the maximum Euclidean distance between two vertices of a facet and only further examine
Obtaining Normal Vectors

To find a normal vector for facet $f$, we first obtain an expression for the hyperplane which contains the facet. For this, we need $K$ points that are on the hyperplane, for which we simply take the first $K$ vertices of the facet. Suppose these points are $a_1, a_2, \ldots, a_K$, then any point $p$ on the hyperplane can be expressed as

$$p = b_f^0 + \sum_{k' = 1}^{K-1} \alpha_{k'} b_{f_{k'}}$$

(2.11)

where $b_f^0 = a_1$, $b_{f_{k'}} = a_{k'+1} - a_1$ are the difference vectors. Starting at point $b_f^0$, any point on the facet-containing hyperplane can be reached by varying the coefficients $\alpha_{k'}$ since the hyperplane is spanned by $b_{f_{k'}}$; $\alpha_{k'}$ are unrestricted in sign and magnitude.

To obtain a vector that is perpendicular to the facet and has unit length, we simply need to find an $n_f \in \mathbb{R}^K$ that is normal to all difference vectors, i.e. $b_f^T n_f = 0$, and satisfies $\|n_f\|_2 = 1$. However, we need the normal vector to point toward the interior of the feasible region. Since the solution for $n_f$ is not unique, we need to check in which direction $n_f$ points and change the sign if it points outward the feasible region.

The procedure for determining the direction of $n_f$ is slightly different for $t = 1$ and $t > 1$. At the first iteration ($t = 1$), the initial facets describe the convex hull around all data points of the subset. Hence, the facet-containing hyperplanes are supporting hyperplanes for the convex hull, i.e. all data points lie on one side of the hyperplane, as illustrated in Figure 2.9. We choose any point $p$ on the facet-containing hyperplane, e.g. $b_f^0$, and a data point $\bar{a}$ that does not lie on the facet. If $n_f$ points toward the interior of the feasible region, the angle formed by $n_f$ and the difference vector $(\bar{a} - p)$ will be less than $90^\circ$, which we can check by simply determining the sign of the inner product $(\bar{a} - p)^T n_f$. If $(\bar{a} - p)^T n_f > 0$, $n_f$ points toward the interior of the feasible region and we set $\bar{n}_f = n_f$; otherwise, $n_f$ points outward and we set $\bar{n}_f = -n_f$.

The procedure described above only applies in the first iteration because the facet-containing hyperplanes in later iterations are not necessarily supporting hyperplanes for the feasible region. However, for $t > 1$, we only have to consider the newly created facets, and we can exploit the fact that we know for each new
2. Data-driven Construction of CRS Models

Figure 2.9: All data points lie on one side of a supporting hyperplane.

facet from which “old” facet it originates. As illustrated in Figure 2.10, the entire old facet $f'$ lies on one side of the hyperplane containing the corresponding new facet $f$. The normal vector of the new facet is supposed to point toward the other side. In order to check the direction in which $n_f$ points, we find a point $p_f$ on the hyperplane containing facet $f$ and a point $p_{f'}$ that lies on facet $f'$ but not on facet $f$. If $n_f$ points into the feasible region, the angle between $n_f$ and $(p_{f'} - p_f)$ will be greater than 90°, i.e. $(p_{f'} - p_f)^T n_f < 0$. In this case, we set $\bar{n}_f = n_f$; otherwise, we set $\bar{n}_f = -n_f$.

Figure 2.10: The dashed line indicates the old facet which led to the two new facets.

Identifying Candidate Points

For each facet $f$, we identify candidate points that can potentially become a new vertex. A candidate point for facet $f$ is defined as follows.

Definition 2.1. Consider a data point $j$ that is not a vertex. Form a polytope with point $j$ as well as the vertices of facet $f$ being vertices of this polytope, which we denote $PT_{fj}$. Point $j$ is a candidate point for facet $f$ if polytope $PT_{fj}$ is completely contained in the current approximation of the feasible region and does not contain any data points in its interior.
In Figure 2.11, a candidate point is illustrated for the top facet of our example. The polytope formed in a two-dimensional case is a triangle. In Figure 2.11a, the triangle formed by the facet and the chosen data point does not contain any points except for the vertices; the chosen data point is therefore a candidate point. In Figure 2.11b, the triangle contains additional data points, but only on the boundary; the corresponding chosen point is therefore also a candidate point. In contrast, the chosen point in Figure 2.11c is not a candidate point because the triangle contains a data point in its interior.

![Diagram](image)

**Figure 2.11:** A candidate point is found if the polytope formed with the facet does not contain any data points in its interior.

For each facet, we have to check for every data point \( j \) that is not a vertex, i.e. \( j \notin V \), if this point is a candidate point. For this, we first solve the LP given in (2.12) for each point \( j \notin V \). The LP can be feasible or infeasible. It is feasible if point \( j \) is on the correct side of the facet, i.e. it can be reached from the facet by moving in the direction of \( \bar{n}_f \). Otherwise, the LP is infeasible. Furthermore, when feasible, the optimal solution of the LP will indicate if there are any data points lying in the interior of polytope \( PT_{fj} \).

\[
\begin{align*}
\min \quad & \sum_{j' \in H^t_j, j' \neq j} \sum_k \left( s_{j'k}^+ + s_{j'k}^- \right) \\
\text{s.t.} \quad & p_k = \sum_{j' \in H^t_j} \nu_{j'} a_{j'k} \quad \forall \ k \\
& \sum_{j' \in H^t_j} \nu_{j'} = 1 \\
& \nu_{j'} \geq 0 \quad \forall \ j' \in H^t_j \\
& a_{jk} = p_k + d \bar{n}_{fk} \quad \forall \ k \\
& d \geq 0
\end{align*}
\]
Eqs. (2.12b)–(2.12f) ensure that the point \( j \) that is being checked is on the side of the facet toward which the normal vector \( \bar{n}_f \) points. In fact, this formulation is slightly more restrictive as it states that point \( j \) has to be reached by moving from a point \( p \) in the direction of \( \bar{n}_f \), where \( p \) is a point on the facet rather than just a point on the facet-containing hyperplane.

Through Eqs. (2.12g)–(2.12j), we check every data point \( j' \) that is not point \( j \) or a vertex of facet \( f \), i.e. \( j' \notin H_f \), \( j' \neq j \), to see if the data point lies in the interior of polytope \( PT_{fj} \). Eqs. (2.12g) and (2.12h) ensure that \( a_{j'} \) is a convex combination of \( a_j \) and \( a_{j''} \) for \( j'' \in H_f \) if the corresponding slack variables are zero. In that case, \( a_{j'} \) is guaranteed to be in the interior of polytope \( PT_{fj} \) because the multipliers are constrained to be greater than or equal to a small value \( \bar{\epsilon} \) in Eq. (2.12i). At the optimal solution of (2.12), \( \sum_k (s_{j'k}^+ + s_{j'k}^-) = 0 \) if point \( j' \) lies in the interior of polytope \( PT_{fj} \), in which case we know that point \( j \) is not a candidate point for facet \( f \).

However, in order to detect the candidate points, only solving (2.12) is not sufficient because we also have to consider the situation illustrated in the example shown in Figure 2.12. Here, point F is the new vertex originating from facet A-B and one can see that the polytope formed by points A, B and F does not contain any data points in its interior. When we now move on to the next facet and examine facet B-C, we see that one can form a polytope with points B, C, and E such that the polytope does not contain any data points in its interior. However, one can clearly see that the formed polytope B-C-E is not fully contained in the current approximation of the feasible region. Therefore, E cannot be a candidate point for facet B-C.

Essentially, we have to check if polytope \( PT_{fj} \) intersects any of the polytopes that have been “cut off” in previous iterations. For instance, in the example in Figure 2.12, we have to check if the polytopes B-C-E and A-B-F overlap each other. This can be achieved by solving the following LP:

\[
\min \sum_{q \in \mathcal{Q}} \sum_k \left( \tilde{s}_{qk}^+ + \tilde{s}_{qk}^- \right) \quad (2.13a)
\]
Figure 2.12: F is the new vertex originating from facet A-B. E is not a candidate point for facet B-C because the resulting polytope B-C-E and the already cut-off polytope A-B-F overlap.

\[ \bar{p}_{j'q} = \mu_{j'q} a_{j'k} + \sum_{j' \in H^f_j} \mu_{j'q} a_{j'k} \quad \forall \ q \in Q, \ k \]  
\hspace{1cm} (2.13b)

\[ \mu_{j'q} + \sum_{j' \in H^f_j} \mu_{j'q} = 1 \quad \forall \ q \in Q \]  
\hspace{1cm} (2.13c)

\[ \mu_{j'q} \geq \bar{\epsilon} \quad \forall \ q \in Q, \ j' = j \text{ or } \in H^f_j \]  
\hspace{1cm} (2.13d)

\[ \bar{p}_{qk} = \sum_{j' \in \bar{H}_q} \gamma_{qj'} a_{j'k} + \hat{s}_{qk}^+ - \hat{s}_{qk}^- \quad \forall \ q \in Q, \ k \]  
\hspace{1cm} (2.13e)

\[ \sum_{j' \in \bar{H}_q} \gamma_{qj'} = 1 \quad \forall \ q \in Q \]  
\hspace{1cm} (2.13f)

\[ \gamma_{qj'} \geq \bar{\epsilon} \quad \forall \ q \in Q, \ j' \in \bar{H}_q \]  
\hspace{1cm} (2.13g)

\[ \hat{s}_{qk}^+, \hat{s}_{qk}^- \geq 0 \quad \forall \ q \in Q, \ k \]  
\hspace{1cm} (2.13h)

where \( Q \) is the set of cut-off polytopes and \( \bar{H}_q \) is the set of vertices of cut-off polytope \( q \). By solving (2.13), we try to find a point \( \bar{p}_q \) for each \( q \in Q \) such that \( \bar{p}_q \) is in the interior of polytope \( PT_{fj} \) and is also in the interior of polytope \( q \). If we can find such a point for at least one of the cut-off polytopes, point \( j \) cannot be a candidate point.

Constraints (2.13b)–(2.13d) force \( \bar{p}_q \) to be a point in the interior of polytope \( PT_{fj} \). Through Eqs. (2.13e)–(2.13h), \( \bar{p}_q \) will also be a point in the interior of polytope \( q \) if the corresponding slack variables are zero. Since we are minimizing the sum of all slack variables, \( \sum_k (\hat{s}_{qk}^+ + \hat{s}_{qk}^-) \) will be driven to zero if polytope \( PT_{fj} \) and polytope \( q \) overlap. Consequently, if point \( j \) has not been ruled out as a potential candidate point after solving (2.12) and at the optimal solution of (2.13),
$\sum_k (s_{qk}^+ + s_{qk}^-) > 0 \quad \forall q \in Q$, we declare point $j$ a candidate point for facet $f$.

Identifying New Vertices and Creating New Facets

With the set of candidate points $C_f$ and the unit-length normal vector $\tilde{n}_f$, we can easily find the candidate point which is furthest away from facet $f$ by solving the following MILP:

\begin{align*}
\text{max} \quad & d \\
\text{s.t.} \quad & p = \sum_{j \in H_f^j} \lambda_j a_j, \quad \sum_{j \in H_f^j} \lambda_j = 1 \quad (2.14a) \\
& \tilde{a} = \sum_{j \in C_f} a_j w_j, \quad \sum_{j \in C_f} w_j = 1 \quad (2.14b) \\
& \tilde{a} = p + d \tilde{n}_f, \quad d \geq 0 \quad (2.14c) \\
& \lambda_j \geq 0, \quad w_j \in \{0, 1\} \quad \forall j \in H_f^j \quad (2.14d)
\end{align*}

where $p$ is a point on facet $f$, as given by the convex combination (2.14b) of the vertices of facet $f$. The binary variable $w_j$ equals 1 if candidate point $a_j$ is chosen. At the optimal solution, $\tilde{a}$ is the candidate point which is furthest away (largest $d$) from facet $f$. Figure 2.13 illustrates the process for one facet.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure2_13.png}
\caption{The data points marked by diamonds are the candidate points for the facet at the top. From all candidate points, the one marked by a hollow diamond is the one furthest away from the facet and is therefore declared as a new vertex.}
\end{figure}

If $d$ is sufficiently large, i.e. $d \geq \delta$, we add the corresponding point to the set of vertices $V$. We construct the new facets by connecting the new vertex with the vertices of each $(K-2)$-dimensional facet of facet $f$. In the two-dimensional case, a facet of the feasible region is a line, and the facets of this facet are always the two end points. Thus, one old facet always leads to two new facets. Note that in higher dimensions, the situation is more complex. The number of new facets created by
using one facet and one new vertex is at least $K$, but could also be significantly larger than $K$. Finally, with the information on the new facets, we update the sets $F$, $\bar{F}$, and $H_{t+1}^f$. Old facets that have been cut off are removed from $F$ while the new facets originated from these old facets are added to $F$. $H_{t+1}^f$, the set of vertices for each $f \in F$ in the next iteration $t+1$, is set accordingly. The set of facets to be examined next, $\bar{F}$, only consists of the new facets.

Figure 2.14 shows the change in the outer approximation of the feasible region at every iteration. As one can see, it requires three major iterations to obtain the final contour of the feasible region.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure2.14.png}
\caption{New facets created at each iteration. The thick red lines show the newly added facets whereas the dashed red lines indicate the facets in the previous iteration from which the new ones originate.}
\end{figure}

\textbf{Remark} At this point, we can write out the CRS model by using the alternative formulation given in Appendix A, which describes the feasible region of the CRS model as the difference of the convex hull and the union of infeasible convex regions. The infeasible convex regions correspond to the set of cut-off polytopes $Q$. However, in order to obtain the convex regions, of which the union forms the feasible region of the CRS model, we require the next step of the algorithm.

\subsection*{2.6.2. Constructing Convex Regions}

With the information on the contour of the feasible region, we can now construct the desired convex regions. Figure 2.15 shows the flowchart for this part of the Phase 2 algorithm, in which we first set an initial number of regions $R$, typically to 1. With $R$ fixed, we can then solve the convex region assignment problem, which assigns vertices $j \in V$, non-vertex points $j' \notin V$ and facets $f \in F$ to $R$ convex regions $r$. If the assignment problem is infeasible, we increase $R$ by 1 and try to solve the problem again.

When the assignment problem is feasible and solved, the algorithm checks whether the obtained convex regions overlap in their interior. If there exist over-
Figure 2.15: Flowchart of the algorithm applied to construct the desired convex regions.

lapping regions, we add cuts to the assignment problem and solve it again. This process is repeated until we reach a feasible solution which yields convex regions that do not overlap. Finally, after removing redundant vertices, we obtain the desired convex regions described by their vertices.

Remark The motivation for disallowing overlapping convex regions is the reduction of redundant feasible space, in our case space belonging to multiple regions. A point belonging to multiple regions corresponds to multiple equivalent solutions and may slow down the optimization algorithm. However, a CRS model with overlapping convex regions is still a valid surrogate model. Therefore, in some cases, eliminating overlapping convex regions may not be necessary.

Convex Region Assignment Formulation

The convex region assignment problem is the centerpiece of the Phase 2 algorithm. The main idea is to partition the approximation of the feasible region defined by the obtained contour into convex regions in the form of polytopes. These polytopes are defined by their vertices, which are chosen from the set of vertices $V$. The union of the convex regions resulting from this partition has to contain all data points.
In order to find these convex regions, we solve an MILP that assigns vertices \( j \in V \), non-vertex points \( j' \notin V \), and facets \( f \in F \) to \( R \) convex regions \( r \). The vertices define the polytopes while the assignment of non-vertex points ensures that all data points are contained in the union of the polytopes. The assignment of facets plays a crucial role in the MILP formulation. Namely, it enables us to formulate constraints through which it can be guaranteed that the regions obtained from the assignment problem are actually polytopes. We elaborate more on this idea later in this section.

**Assignment of vertices and facets**  The following constraints assign vertices and facets to regions.

\[
\begin{align*}
\sum_r x_{rf} &= 1 \quad \forall f \\
\sum_f x_{rf} &\geq 1 \quad \forall r \\
-x_{rf} + \sum_{f' \prec f} x_{r-1,f'} &\leq 0 \quad \forall r > 1, f \\
x_{rf} &\leq y_{rj} \quad \forall r, f, j \in H_f \\
\sum_{j \in V} y_{rj} &\geq K + 1 \quad \forall r \\
\sum_r y_{rj} &\leq 1 + M w_j \quad \forall j \in V \\
w_j, x_{rf}, y_{rj} &\in \{0,1\} \quad \forall r, f, j \in V
\end{align*}
\]

where the binary variable \( x_{rf} \) equals 1 if facet \( f \) is assigned to region \( r \), \( y_{rj} \) equals 1 if vertex \( j \) is assigned to region \( r \), and \( w_j \) equals 1 if vertex \( j \) is assigned to more than one region.

Eq. (2.15a) states that a facet can only be assigned to one region, while constraint (2.15b) ensures that at least one facet is assigned to a region. Constraint (2.15c) is a symmetry-breaking constraint that enforces facet-to-region assignment in lexicographic order. Constraint (2.15d) states that if facet \( f \) is assigned to region \( r \), all vertices of facet \( f \), i.e. all \( j \in H_f \), are also assigned to region \( r \). To encourage the creation of regions that fill \( K \)-dimensional volumes, the minimum number of vertices assigned to a region is set to \( K + 1 \) by constraint (2.15e). Furthermore, constraint (2.15f) states that a vertex \( j \) can only be assigned to more than one region if \( w_j \) equals 1.
Assignment of non-vertex points The following constraints assign non-vertex points to regions such that the points are inside the regions to which they are assigned.

\[ \sum_r z_{rj} = 1 \quad \forall \ j \notin V \quad (2.16a) \]

\[ a_j = \sum_{j' \in V} \lambda_{rj'j} a_{j'} + s^+_{rj} - s^-_{rj} \quad \forall \ r, j \notin V \quad (2.16b) \]

\[ \lambda_{rj'j} \leq y_{rj'} \quad \forall \ r, j' \in V, j \notin V \quad (2.16c) \]

\[ \sum_{j' \in V} \lambda_{rj'j} = 1 \quad \forall \ r, j \notin V \quad (2.16d) \]

\[ s^+_{rj} \leq M (1 - z_{rj}) \quad \forall \ r, j \notin V \quad (2.16e) \]

\[ s^-_{rj} \leq M (1 - z_{rj}) \quad \forall \ r, j \notin V \quad (2.16f) \]

\[ \lambda_{rj'j}, s^+_{rj}, s^-_{rj} \geq 0 \quad \forall \ r, j' \in V, j \notin V \quad (2.16g) \]

\[ z_{rj} \in \{0, 1\} \quad \forall \ r, j \notin V \quad (2.16h) \]

where the binary variable \( z_{rj} \) equals 1 if non-vertex point \( j \) is assigned to region \( r \).

Eq. (2.16a) enforces that a non-vertex point \( j \) can only be assigned to one region. Constraints (2.16b)–(2.16g) ensure that point \( j \) is inside the convex region \( r \) if \( j \) is assigned to \( r \). The idea is that the corresponding slack variables become zero if \( z_{rj} \) equals 1 so that \( a_j \) is constrained to be a convex combination of the vertices of region \( r \).

Constraining regions to be polytopes In the convex region assignment, we have to ensure that the obtained regions are in fact polytopes. In order to formulate such constraints, we make full use of the information from the contour of the feasible region that we obtained in the previous part of the Phase 2 algorithm. The idea is to force all vertices assigned to region \( r \) to be on the same side of the hyperplane containing facet \( f \) for every facet \( f \) that is assigned to region \( r \). To illustrate this idea, we label the vertices in our illustrative example as shown in Figure 2.16.

Suppose we assign the facets C-D, D-E, E-F, F-G and the corresponding vertices C, D, E, F, G to the same region. Clearly, they form a convex region as shown in Figure 2.17a. One can see that each facet-containing hyperplane is a supporting hyperplane for the formed convex region, i.e. all points in the region lie on the same side of the hyperplane. If we then want to also include facet B-C and vertex B in the region, the region is not convex anymore, as shown in Figure 2.17b. Here, both hyperplanes corresponding to facet B-C and C-D cut through the region. Therefore,
2. DATA-DRIVEN CONSTRUCTION OF CRS MODELS

Figure 2.16: Contour of the feasible region with labeled vertices.

Figure 2.17: A region is convex if and only if the hyperplanes containing the assigned facets are supporting hyperplanes for the region.

this solution is infeasible for the convex region assignment problem.

Since each convex region is defined by its vertices, we only have to constrain all vertices assigned to the region to be on the same side of the corresponding facet-containing hyperplanes. For a point $a$ to be on the side of the hyperplane containing facet $f$ in which the normal vector $\bar{n}_f$ points, the inner product between the difference vector $(a - p)$, where $p$ is a point on the hyperplane, and $\bar{n}_f$ has to be greater than or equal to zero. This results in the following constraint:

$$\left( a_j - b_{jf}^0 \right)^T \bar{n}_f \geq -M \left( 2 - x_{rf} - y_{rj} \right) \ \forall \ r, f, j \in V$$

(2.17)

which states that $\left( a_j - b_{jf}^0 \right)^T \bar{n}_f \geq 0$ has to hold for all pairs of vertex $j$ and facet $f$ that are assigned to the same region $r$. Here, $b_{jf}^0$ is a vertex of facet $f$. 
Objective function  The MILP minimizes

\[ \phi = \sum_r \sum_{j \in V} y_{rj} + \sum_{j \in V} w_j \]  

(2.18)

which is the total number of vertex-to-region assignments and the total number of vertices that are assigned to more than one region. This is a heuristic used to discourage the construction of overlapping convex regions.

The full convex region assignment problem is then:

\[
\begin{align*}
\text{min} & \quad \phi \\
\text{s.t.} & \quad \text{Eqs. (2.15)--(2.18)}
\end{align*}
\]  

(2.19)

Identifying and Eliminating Overlapping Convex Regions

The optimal solution of the convex region assignment problem may yield convex regions that overlap each other in their interior, as illustrated in the first sketch in Figure 2.18. To identify overlapping regions, we solve an LP that is almost identical to (2.9) used in Phase 1. Except, here we restrict the multipliers to be greater than a small value \( \bar{\epsilon} \) because we want to detect regions which overlap in their interior. Regions that only overlap at the boundary are not considered overlapping regions.

![Figure 2.18: The convex region assignment problem may yield overlapping convex regions. By adding the proposed cuts, these solutions can be avoided.](image)

If overlapping regions exist, we can use the solution of the LP to generate cuts equivalent to the ones in (2.10) described in Section 2.5.3 and add them to the convex region assignment problem. By repeatedly resolving (2.19) with added cuts if overlapping regions are detected, we converge to a solution with no overlap, such as the one illustrated in the second sketch of Figure 2.18.
Removing Redundant Vertices

The algorithm may declare data points vertices although they may later be assigned to convex regions in which they are not vertices of the formed polytopes. In the example shown in Figure 2.19, point B has been declared a vertex in the first part of Phase 2. However, by solving the convex region assignment problem, we find the two polytopes shown in Figure 2.19b which provide an accurate approximation of the feasible region. Point B is on a facet of one of the two polytopes but is not a vertex. Hence, point B is redundant because it is not needed for describing the polytope.

![Diagram showing vertices and polytopes](image)

**Figure 2.19:** In this example, point B is a redundant vertex because it has been declared a vertex, but is not a vertex of any of the constructed convex regions.

We can eliminate these redundant vertices by finding the vertices of each polytope and removing the ones that are not true vertices. The algorithm terminates after this step since now we have obtained all convex regions with the corresponding vertices required to build the CRS model.

### 2.7. Summary of the Algorithm

To summarize, we show the complete CRS algorithm at once. Note that for the description of the Phase 2 algorithm, we reintroduce the subset index $i$, which we omitted for the sake of readability in Section 2.6.
Phase 1 (Subset assignment with cost correlation constraints)

Step 1.1 (Initialization) Set iteration counter \( t = 1 \), initial number of subsets \( m = 1 \), and specify error tolerance \( \epsilon \).

Step 1.2 (Subset assignment) Construct set of subsets \( I^t = \{1, \ldots, m\} \), and solve (2.6) or (2.7) to assign data points to subsets. If feasible, construct the sets of assigned data points \( J_i \), and go to Step 1.3. Otherwise, set \( t = t + 1, m = m + 1 \), reset the set of overlap elimination cuts \( S = \emptyset \), and repeat Step 1.2.

Step 1.3 (Constructing convex hulls) Apply the Quickhull algorithm to each subset \( i \in I^t \) to obtain the sets of vertices \( V_i \) for the convex hulls around the data points assigned to each subset.

Step 1.4 (Identifying overlapping convex hulls and generating cuts) Check for overlapping convex hulls by solving (2.9). If no overlap, construct sets of facets \( F_i \) and sets of facet vertices \( H_{ij} \), and go to Step 2.1. Otherwise, generate cuts as given by (2.10), add the cuts to (2.6) or (2.7), and return to Step 1.2.

Phase 2 (Convex region assignment)

Step 2.1 (Starting loop over set of subsets) Set subset index \( i = 1 \), specify minimum distances \( \delta \).

Step 2.2 (Initializing contour construction) Set iteration counter \( t = 1 \), the set of facets to be examined \( \bar{F} = F_i \), and the set of facet vertices \( H_{ij} = H_{ij} \).

Step 2.3 (Construction of contour of feasible region)

Step 2.3.1 (Starting loop over set of facets) Set \( f \) to be the first element in \( \bar{F} \).

Step 2.3.2 (Obtaining normal vector) Obtain a vector \( n_f \) normal to facet \( f \) as described in Section 2.6.1.1. Set \( \bar{n}_f = n_f \) if \( n_f \) points toward the interior of the feasible region, set \( \bar{n}_f = -n_f \) if otherwise.

Step 2.3.3 (Finding candidate points for new vertex)

Step 2.3.4a Set \( j \) to be the first element of \( P = \{ j : j \in J_i \land j \notin V_i \} \).

Step 2.3.4b Solve (2.12). If feasible and no point detected in the interior of polytope \( PT_{fj} \), go to Step 2.3.4c. Otherwise, go to Step 2.3.4d.

Step 2.3.4c Solve (2.13). If no overlap between polytope \( PT_{fj} \) and any cut-off polytopes \( q \in Q \), add \( j \) to \( C \).
Step 2.3.4d If \( j \) is the last element in \( P \), go to Step 2.3.5. Otherwise, set \( j \) to the next element in \( P \) and return to Step 2.3.4b.

Step 2.3.5 (Determining new vertex and facets) If \( C \neq \emptyset \), solve (2.14), obtain distance \( d \) and the corresponding point \( j \). If \( d \geq \delta \), add point \( j \) to the set of vertices \( V_i \), create new facets, remove facet \( f \) from and add new facets to \( F_i \), add cut-off polytope to \( Q \) and update set of cut-off polytope vertices \( \bar{H}_{iq} \).

Step 2.3.6 (Closing loop over set of facets) If \( f \) is the last element in \( \bar{F} \), go to Step 2.3.7. Otherwise, set \( f \) to the next element in \( \bar{F} \) and return to Step 2.3.2.

Step 2.3.7 (Updating contour information) Construct \( H^{t+1}_{ij} \) according to \( F_i \). If new facets have been created, empty \( \bar{F} \) and add the new facets to it, set \( t = t + 1 \), and return to Step 2.3.1. Otherwise, go to Step 2.4.

Step 2.4 (Initializing convex region assignment) Set initial number of regions \( R = 1 \).

Step 2.5 (Convex region assignment) Solve convex region assignment problem (2.19). If feasible, go to Step 2.6. Otherwise, set \( R = R + 1 \), reset the set of overlap elimination cuts \( L = \emptyset \), and repeat Step 2.5.

Step 2.6 (Identifying overlapping regions and generating cuts) Check for overlapping convex regions. If no overlap, go to Step 2.7. Otherwise, generate cuts, add the cuts to (2.19), and return to Step 2.5.

Step 2.7 (Removing redundant vertices) Remove redundant vertices as described in Section 2.6.2.3. If \( i = m \), stop and report the solution in form of the parameter values \( VT_{irjk} \) corresponding to subset \( i \), convex region \( r \), vertex \( j \), and dimension \( k \). If \( i < m \), set \( i = i + 1 \) and return to Step 2.2.

The algorithm is implemented using MATLAB R2012a (The Mathworks Inc., 2012) and GAMS 24.2.1 (GAMS Development Corporation, 2013).

2.8. Numerical Results

In the following, we test the proposed algorithm in an extensive computational study as well as in a real-world industrial case study. In all instances, the algorithm was executed on an Intel® Core™ i7-2600 machine at 3.40 GHz with 8 processors and 8 GB RAM running Windows 7 Professional.
2.8.1. Computational Study

We perform a computational study on the two parts of the algorithm, Phase 1 and Phase 2. To assess the algorithm’s computational performance, we apply it to various instances and demonstrate the impact of the number of data points, the dimensionality, and the level of nonlinearities and nonconvexities on the performance of the algorithm.

It should be emphasized that like most surrogate models, the CRS model is constructed “offline”. Once it is created, it can be used “online” (e.g. in a scheduling model) and only has to be updated if the process characteristics have changed significantly. Hence, it is typically acceptable for the computation time required for the construction of a surrogate model to be relatively large since it usually only has to be done once.

Phase 1

Phase 1 strongly depends on the level of nonlinearity in the cost function that we want to approximate. In practice, it is only suited for cases in which only a small number of linear functions are required to approximate the true cost function with the specified maximum fitting error. Obviously, the more nonlinear the true cost function is, the more subsets the algorithm will create. In the worst case, if the true cost function is extremely nonlinear, we may obtain one subset for each individual data point, which makes the construction of a CRS model meaningless.

In the following, we consider constructed instances in 2D and 3D, in which we can clearly see the structure of the cost function in relation to the feasible region, and observe how the algorithm performs in these different instances. Each case is denoted by four numerals, e.g. “2-221-3-A”, where the first number states the dimensionality, the second number is the number of data points, the third number indicates the number of different linear cost functions in the given data set, and the last part of the name distinguishes different variants.

For each dimensionality, we generate four pairs of instances. Each pair consists of two instances that have the same geometric structure but differ in the number of data points. For 2D, we have four instances with 121 data points in each case and the corresponding four instances with 221 data points each. Figure 2.20 shows the final subset assignment solutions for the four 2D cases with 221 data points. The different colors of the regions indicate different cost correlations.

The computational results for the eight 2D cases are shown in Table 2.1, which
lists the number of subset assignment problems solved, the number of subsets, and the wall-clock time in seconds. The results clearly show that the computation time strongly depends on the number of subset assignment problems that have to be solved to obtain the final result. Naturally, larger data sets increase the size of the subset assignment problem and therefore the required computing time. However, the relationship between the cost correlation and the location of the data points seems to have an even greater impact on the algorithm, more specifically on the number of assignment problems that need to be solved. Essentially, the more feasible solutions we have that lead to overlapping convex hulls, the more iterations are required, which significantly increases the computation time. A good example is Case 2-221-3-B, which only involves three different linear cost correlations but results in five subsets due to the locations of the data points that do not allow three subsets with non-overlapping convex hulls.

Table 2.1: Computational results from Phase 1 for the 2D cases

<table>
<thead>
<tr>
<th>Case</th>
<th># of Subset Assignment Problems Solved</th>
<th># of Subsets</th>
<th>Wall-Clock Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-121-2-A</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2-121-2-B</td>
<td>5</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>2-121-3-A</td>
<td>49</td>
<td>4</td>
<td>635</td>
</tr>
<tr>
<td>2-121-3-B</td>
<td>24</td>
<td>5</td>
<td>163</td>
</tr>
<tr>
<td>2-221-2-A</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2-221-2-B</td>
<td>5</td>
<td>3</td>
<td>70</td>
</tr>
<tr>
<td>2-221-3-A</td>
<td>22</td>
<td>4</td>
<td>2164</td>
</tr>
<tr>
<td>2-221-3-B</td>
<td>83</td>
<td>5</td>
<td>4111</td>
</tr>
</tbody>
</table>

Figure 2.21 shows the 3D cases with 214 data points, and the computational
results for all eight 3D cases are presented in Table 2.2. Here, we make similar observations as in the 2D cases. Among the considered instances, Cases 3-120-3-B and 3-214-3-B stand out. The data in these two instances represent the same geometric structure; however, although we have more data points in Case 3-214-3-B, the computation time is shorter due to the significantly smaller number of subset assignment problems that have to be solved. In this particular case, we observe the phenomenon that the larger data set “guides” the algorithm more quickly toward a solution with no overlapping convex hulls.

![Subset assignments resulting from Phase 1 for the 3D cases with 214 data points.](image)

**Figure 2.21:** Subset assignments resulting from Phase 1 for the 3D cases with 214 data points.

<table>
<thead>
<tr>
<th>Case</th>
<th># of Subset Assignment Problems Solved</th>
<th># of Subsets</th>
<th>Wall-Clock Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-120-2-A</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3-120-2-B</td>
<td>5</td>
<td>3</td>
<td>20</td>
</tr>
<tr>
<td>3-120-3-A</td>
<td>6</td>
<td>4</td>
<td>30</td>
</tr>
<tr>
<td>3-120-3-B</td>
<td>88</td>
<td>4</td>
<td>2655</td>
</tr>
<tr>
<td>3-214-2-A</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3-214-2-B</td>
<td>7</td>
<td>3</td>
<td>340</td>
</tr>
<tr>
<td>3-214-3-A</td>
<td>6</td>
<td>4</td>
<td>145</td>
</tr>
<tr>
<td>3-214-3-B</td>
<td>10</td>
<td>4</td>
<td>650</td>
</tr>
</tbody>
</table>

**Table 2.2:** Computational results from Phase 1 for the 3D cases

**Phase 2**

Independent from Phase 1, we analyze the performance of the Phase 2 algorithm by applying it to various data sets in 2D, 3D, and 4D. Since we want to investigate the impact of different factors, such as the number of data points, the prespecified tolerance, and the geometry of the feasible space, some common structure in the cases has to be maintained in order to make proper comparisons. Therefore, the
data are partially randomized but not chosen completely randomly. Each case is named by using two or three numerals, e.g. “2-167-A1”, where the first number indicates the dimensionality of the given data, the second number is the number of data points, and the third part of the name further distinguishes different variants.

Figures 2.22 and 2.23 show the cases in 2D. We consider the two base cases shown in Figure 2.22a and 2.23a, in which the data points are aligned on a regular grid. As one can see, these two cases differ in the number of data points but have the same geometric structure. All the other cases shown in Figures 2.22 and 2.23 are created by randomly perturbing the data points in the corresponding base cases. The random cases shown in the first rows, and the ones in the second rows use the same data but the algorithm has been applied to the latter with a larger tolerance \( \delta \). It is clear from the shown results that the convex regions obtained from the algorithm can vary significantly when different \( \delta \) values are chosen (e.g. compare Cases 2-91-C1 and 2-91-C2). In general, the number of regions decreases with a greater \( \delta \).

![Figure 2.22: Convex regions resulting from Phase 2 for the 2D cases with 91 data points.](image)

Table 2.3 summarizes the computational results for the 2D cases. For each case, we report the following statistics:

- prespecified tolerance \( \delta \);
Figure 2.23: Convex regions resulting from Phase 2 for the 2D cases with 167 data points.

- initial number of facets, which are the facets of the initial convex hull around all data points;
- final number of facets, which are the facets created to describe the contour;
- number of $t$-iterations, which are the iterations in which new facets are created;
- number of convex region assignment (CRA) problems solved;
- wall-clock time in seconds.

From the results, we can see that larger $\delta$ values lead to smaller numbers of facets created by the algorithm, which typically result in smaller numbers of convex regions. The computation time significantly increases with the number of data points since it increases the time required for each $t$-iteration and makes the CRA problem considerably more difficult to solve. Equally important is the impact of the level of nonconvexity of the feasible region, which is reflected in the number new facets that have to be created. A larger number of facets increases the complexity of the CRA problem and the likelihood that the CRA problem has to be resolved several times because of overlapping convex regions.
Table 2.3: Computational results from Phase 2 for the 2D cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>δ</th>
<th>Initial # of Facets</th>
<th>Final # of Facets</th>
<th># of t-Iterations</th>
<th># of Regions</th>
<th># of CRAs Solved</th>
<th>Wall-Clock Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-91</td>
<td>0.05</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>51</td>
</tr>
<tr>
<td>2-91-A1</td>
<td>0.05</td>
<td>15</td>
<td>17</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>352</td>
</tr>
<tr>
<td>2-91-A2</td>
<td>0.1</td>
<td>15</td>
<td>17</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>350</td>
</tr>
<tr>
<td>2-91-B1</td>
<td>0.05</td>
<td>12</td>
<td>16</td>
<td>3</td>
<td>3</td>
<td>5</td>
<td>356</td>
</tr>
<tr>
<td>2-91-B2</td>
<td>0.1</td>
<td>12</td>
<td>15</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>325</td>
</tr>
<tr>
<td>2-91-C1</td>
<td>0.05</td>
<td>12</td>
<td>21</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>998</td>
</tr>
<tr>
<td>2-91-C2</td>
<td>0.1</td>
<td>12</td>
<td>15</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>425</td>
</tr>
<tr>
<td>2-167</td>
<td>0.05</td>
<td>8</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>109</td>
</tr>
<tr>
<td>2-167-A1</td>
<td>0.05</td>
<td>16</td>
<td>19</td>
<td>3</td>
<td>3</td>
<td>7</td>
<td>1793</td>
</tr>
<tr>
<td>2-167-A2</td>
<td>0.1</td>
<td>16</td>
<td>18</td>
<td>3</td>
<td>3</td>
<td>8</td>
<td>1245</td>
</tr>
<tr>
<td>2-167-B1</td>
<td>0.05</td>
<td>17</td>
<td>21</td>
<td>3</td>
<td>4</td>
<td>8</td>
<td>1842</td>
</tr>
<tr>
<td>2-167-B2</td>
<td>0.1</td>
<td>17</td>
<td>19</td>
<td>2</td>
<td>3</td>
<td>12</td>
<td>1052</td>
</tr>
<tr>
<td>2-167-C1</td>
<td>0.05</td>
<td>15</td>
<td>20</td>
<td>4</td>
<td>4</td>
<td>17</td>
<td>4422</td>
</tr>
<tr>
<td>2-167-C2</td>
<td>0.1</td>
<td>15</td>
<td>17</td>
<td>2</td>
<td>3</td>
<td>11</td>
<td>1565</td>
</tr>
</tbody>
</table>

Figures 2.24 and 2.25 show the 3D cases considered in this computational study. Similar to the 2D instances, we have two base cases shown in Figures 2.24a and 2.25a that differ in the number of data points but not the geometric structure. The remaining cases are generated by randomly perturbing subsets of the data points from the base cases. In order to avoid creating excessively large numbers of facets, the number of $t$-iterations has been limited to 3. Also, we allow overlapping convex regions, i.e. the CRA problem does not have to be resolved when convex regions overlap each other. Hence, the number of CRA problems solved is equal to the number of convex regions obtained.

Table 2.4 summarizes the computational results for the 3D cases. In general, the same observations can be made as in the 2D cases. However, when comparing the 2D with the 3D instances, we can see a stark increase in computational complexity resulting from the increase in dimensionality. The computation time has grown dramatically despite limiting the number of $t$-iterations and allowing overlapping convex regions. This is not only a result of the additional dimension that increases the size of the various optimization problems, but it also stems from the fact that the approximation of nonconvex higher-dimensional spaces generally requires more convex regions.
2. Data-driven Construction of CRS Models

Unlike Phase 1, which clearly also applies to dimensions higher than three, Phase 2 is more involved and its applicability to higher-dimensional data may not be obvious. Therefore, although the results cannot be easily visualized, we apply the algorithm to some pseudo-randomly generated instances with data in 4D. Ta-

Table 2.4: Computational results from Phase 2 for the 3D cases.

<table>
<thead>
<tr>
<th>Case</th>
<th>$\delta$</th>
<th>Initial # of Facets</th>
<th>Final # of Facets</th>
<th># of $t$-Iterations</th>
<th># of Regions</th>
<th>Wall-Clock Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3-114</td>
<td>0.1</td>
<td>7</td>
<td>12</td>
<td>2</td>
<td>4</td>
<td>116</td>
</tr>
<tr>
<td>3-114-A1</td>
<td>0.1</td>
<td>21</td>
<td>35</td>
<td>3</td>
<td>13</td>
<td>3405</td>
</tr>
<tr>
<td>3-114-A2</td>
<td>0.2</td>
<td>21</td>
<td>25</td>
<td>2</td>
<td>6</td>
<td>548</td>
</tr>
<tr>
<td>3-114-B1</td>
<td>0.1</td>
<td>24</td>
<td>42</td>
<td>3</td>
<td>19</td>
<td>5754</td>
</tr>
<tr>
<td>3-114-B2</td>
<td>0.2</td>
<td>24</td>
<td>32</td>
<td>2</td>
<td>9</td>
<td>1302</td>
</tr>
<tr>
<td>3-114-C1</td>
<td>0.1</td>
<td>23</td>
<td>40</td>
<td>3</td>
<td>20</td>
<td>5866</td>
</tr>
<tr>
<td>3-114-C2</td>
<td>0.2</td>
<td>23</td>
<td>35</td>
<td>2</td>
<td>14</td>
<td>3700</td>
</tr>
<tr>
<td>3-234</td>
<td>0.1</td>
<td>7</td>
<td>12</td>
<td>2</td>
<td>4</td>
<td>210</td>
</tr>
<tr>
<td>3-234-A1</td>
<td>0.1</td>
<td>17</td>
<td>24</td>
<td>2</td>
<td>6</td>
<td>1493</td>
</tr>
<tr>
<td>3-234-A2</td>
<td>0.2</td>
<td>17</td>
<td>22</td>
<td>2</td>
<td>4</td>
<td>962</td>
</tr>
<tr>
<td>3-234-B1</td>
<td>0.1</td>
<td>19</td>
<td>36</td>
<td>3</td>
<td>36</td>
<td>20,991</td>
</tr>
<tr>
<td>3-234-B2</td>
<td>0.2</td>
<td>19</td>
<td>28</td>
<td>1</td>
<td>13</td>
<td>6976</td>
</tr>
<tr>
<td>3-234-C1</td>
<td>0.1</td>
<td>18</td>
<td>37</td>
<td>3</td>
<td>37</td>
<td>21,853</td>
</tr>
<tr>
<td>3-234-C2</td>
<td>0.2</td>
<td>18</td>
<td>27</td>
<td>2</td>
<td>11</td>
<td>4269</td>
</tr>
</tbody>
</table>

Figure 2.24: Convex regions resulting from Phase 2 for the 3D cases with 114 data points
Figure 2.25: Convex regions resulting from Phase 2 for the 3D cases with 234 data points

Table 2.5: Computational results from Phase 2 for the 4D cases

<table>
<thead>
<tr>
<th>Case</th>
<th>Initial # of Facets</th>
<th>Final # of Facets</th>
<th># of t-Iterations</th>
<th># of Regions</th>
<th>Wall-Clock Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>4-81-A</td>
<td>24</td>
<td>28</td>
<td>1</td>
<td>8</td>
<td>451</td>
</tr>
<tr>
<td>4-81-B</td>
<td>28</td>
<td>40</td>
<td>1</td>
<td>14</td>
<td>2337</td>
</tr>
<tr>
<td>4-81-C</td>
<td>43</td>
<td>61</td>
<td>2</td>
<td>16</td>
<td>4467</td>
</tr>
<tr>
<td>4-256-A</td>
<td>27</td>
<td>34</td>
<td>1</td>
<td>10</td>
<td>6487</td>
</tr>
<tr>
<td>4-256-B</td>
<td>33</td>
<td>47</td>
<td>2</td>
<td>16</td>
<td>15,193</td>
</tr>
<tr>
<td>4-256-C</td>
<td>38</td>
<td>52</td>
<td>2</td>
<td>18</td>
<td>16,989</td>
</tr>
</tbody>
</table>

The main insight drawn from this computational study is that besides the more obvious factors such as dimensionality and number of data points, the performance of the algorithm (Phase 1 as well as Phase 2) strongly depends on the level of nonlinearity and nonconvexity implied by the data set. The computational effort significantly increases with the geometric complexity since the approximation of a more complex geometric structure generally requires a larger number of convex
2.8.2. Industrial Case Study

We apply the proposed algorithm to a real-world case study provided by Praxair. Note that this set of data and the resulting CRS model are not the ones used in the actual industrial application, which cannot be shown here due to confidentiality reasons. With this rather unfavorable set of data, however, we mainly want to demonstrate the effect of overfitting and show the importance of effective data collection.

The objective is to construct a CRS model of a production process that can be integrated into a scheduling optimization problem, which is formulated as an MILP. The variable space of the CRS model has two dimensions since we are only interested in the production rates of Products A and B. The cost function is the power consumption.

The data were drawn from the actual process for the purpose of analyzing the extent of the feasible operational region (not with the construction of a CRS model in mind). Since sampling from such a complex production process is expensive, only 55 data points were taken. The attempt was made to sample at the boundaries of the feasible region. Each data point carries production rates for Products A and B as well as the corresponding power consumption value. The data are plotted in Figure 2.26a. Note that for this case study, we use normalized values.

\[\epsilon\]

Figure 2.26: In Phase 1, the given data points are partitioned into two disjoint subsets, denoted subsets 1 and 2.

For the subset assignment in Phase 1, we set the maximum error tolerance $\epsilon$
to 3%. This results in the partitioning of the given data points into two disjoint convex hulls shown in Figure 2.26b. The cost constants and coefficients for the linear power consumption correlations are shown in Table 2.6.

Table 2.6: Constants and coefficients for the power consumption correlations.

<table>
<thead>
<tr>
<th>Subset</th>
<th>( b )</th>
<th>( c_A )</th>
<th>( c_B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.900</td>
<td>0.062</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>0.703</td>
<td>0.127</td>
<td>0.236</td>
</tr>
</tbody>
</table>

As observed in the computational study, the result from Phase 2 strongly depends on the specified \( \delta \), the minimum distance between a new vertex and the facet that it originates from. For comparison, we apply the Phase 2 algorithm for four different \( \delta \)s, for which the resulting convex regions are shown in Figure 2.27. One can see that the smaller \( \delta \) is, the more points tend to be declared vertices, which leads to the construction of more convex regions. While for \( \delta = 0.10 \), 4 convex regions are constructed, we obtain 11 convex regions for \( \delta = 0.04 \). For \( \delta = 0.08 \) and \( \delta = 0.06 \), we have the same number of convex regions. However, the convex regions for \( \delta = 0.06 \) describe a tighter envelope around the data points.

This example illustrates a key principle in data-driven approaches: A good method does not necessarily lead to good results if it is not given a sufficiently large set of useful data. Clearly, among the cases shown in Figure 2.27, \( \delta = 0.04 \) results in the tightest envelope around all data points. However, this may not provide the most accurate approximation of the feasible region. Especially when observing the shape of the union of the convex regions for Subset 2, there is reason to suspect that this does not resemble the actual feasible region of the process. However, if we had more data points filling up a larger portion of the feasible space, we might obtain a different result or be more confident about the obtained result. Due to the limited number of data points, one would probably rather apply the CRS model resulting from \( \delta > 0.04 \), e.g. \( \delta = 0.06 \). In practice, one should try to validate the surrogate model, e.g. by drawing a set of additional data points from the process to test them for feasibility in the surrogate model, and vice versa.

2.9. Remarks

In the following, we provide some remarks on the proposed methodology and limiting features of the algorithm, which will motivate future work:
2. DATA-DRIVEN CONSTRUCTION OF CRS MODELS

Figure 2.27: Depending on $\delta$, the Phase 2 algorithm constructs different convex regions.

- The curse of dimensionality is a major concern when applying the proposed approach. The higher the dimensionality of the data, the more data points are required to accurately represent the feasible space. Of course, the more data points are involved, the higher is also the computational complexity, which increases up to a point when it becomes intractable. Also, approximating nonconvexities in higher dimensions generally requires more convex regions. At some point, so many convex regions will be required such that the advantage of using a CRS model compared to a nonlinear model diminishes. For practical purposes, this approach may be applicable to data with up to 5 or 6 dimensions. We should note, however, that in practice the computational expense for constructing the surrogate model is not so relevant as this computation is performed only once. What is more relevant is having
the surrogate model itself as it can then be applied extensively at much lower cost compared, for instance, with a complex nonlinear model.

- In Phase 1, we obtain subsets of data points that are disjoint, which leads to regions that are disconnected. The algorithm is designed this way because continuity of the piecewise linear approximation of the cost function at the boundaries of the regions cannot be guaranteed. However, especially when the number of data points is small, this may lead to overly large gaps between the regions.

- In Phase 2, we assume that the feasible region to be approximated is simply connected. However, it is possible that the union of the convex regions resulting from the algorithm is not simply connected, i.e. it contains empty spaces. In that case, we could extend the algorithm and add a mechanism to detect and fill such empty spaces. Nonetheless, the simple connectedness assumption may not always be valid, which is not taken into account in the proposed algorithm.

- In Phase 2, a facet will not be further examined if it contains a point in its relative interior since this point would be cut off if the vertices of the facet were used to create new facets. However, this might terminate the facet generation procedure prematurely. The point in the facet’s interior could actually be a vertex of the contour of the feasible region, which could be used to further refine the approximation. Detecting such points and integrating them in the procedure, however, is a challenge, especially in high-dimensional cases.

### 2.10. Summary

The work presented in this chapter has addressed the challenge of generating accurate and computationally efficient surrogate models for mixed-integer programming frameworks. We have introduced the idea of Convex Region Surrogate models, which can be formulated as sets of mixed-integer linear constraints that provide good approximations of nonlinearities and nonconvexities. In a CRS model, the feasible region is approximated by the union of convex regions in the form of polytopes. Furthermore, for each region, the cost function is approximated by a linear function.

We have presented a two-phase algorithm for the construction of CRS models based on given data. In Phase 1, the algorithm partitions the data points into
disjoint subsets such that a linear cost correlation can be obtained within an error
tolerance for all points in each subset, and that the convex hulls around the points
of each subset do not overlap. In Phase 2, convex regions are constructed for each
subset such that the union of the convex regions describes an accurate approxima-
tion of the feasible region.

An extensive computational study has been conducted in order to assess the
performance of the proposed algorithm on different data sets. We have further ap-
plied the methodology to an industrial test case drawn from a Praxair plant. Specif-
ically, data from a real production process have been used to generate a CRS model
with a feasible region in the product space and power consumption as the cost
function. The case studies have demonstrated the applicability of the algorithm as
well as provided insights into the impact of given data on the performance of the
algorithm.
Having developed a general framework for generating computationally efficient surrogate process models in Chapter 2, we are now equipped to consider the higher-level scheduling problem. In this chapter, we develop a general discrete-time scheduling model for continuous power-intensive process networks. The proposed mode-based model captures all critical operational constraints and considers various types of power contracts from which electricity can be purchased. The resulting MILP formulation has proven to be computationally efficient and is a basic component of all models presented in Chapters 4–7.

This chapter is organized as follows. Section 3.1 provides a brief literature review on production scheduling of continuous processes and related works considering varying electricity prices. The problem statement is presented in Section 3.2 before the MILP model is developed in Section 3.3. In Section 3.4, the proposed model is applied to an illustrative example as well as an industrial air separation case study. In Section 3.5, we close the chapter with a summary of the results.

3.1. Background

Production scheduling has been an active area of research in PSE since the late 1970s. Since then, much progress has been made in the modeling of both batch and continuous scheduling problems as well as in the development of efficient methods for solving these models (Méndez et al., 2006; Maravelias, 2012). In this work, we consider continuous production processes, for which numerous production scheduling models have been proposed in the literature. Some of the proposed models are based on the STN framework; see, for example, Ierapetritou & Floudas (1998), Giannelos & Georgiadis (2002), and Shaik & Floudas (2007). Many other
models are based on the RTN framework, e.g. Zhang & Sargent (1996), Schilling & Pantelides (1996), and Castro et al. (2004). Some researchers have combined batch and continuous scheduling, or have focused on the case with parallel processes (Karimi & McDonald, 1997; Kopanos et al., 2011). Also, cyclic scheduling has been considered (Sahinidis & Grossmann, 1991; Pinto & Grossmann, 1994; Alle et al., 2004). Note that many of the recent research efforts have focused on the development of new continuous-time formulations.

Castro et al. (2009, 2011) present RTN-based discrete-time and continuous-time models for the scheduling of continuous plants under variable electricity cost. Computational experiments show that the discrete-time models are significantly more efficient than the continuous-time models when solving large-scale instances. Mitra et al. (2012a) have developed a discrete-time scheduling model based on the concept of operating modes, which allows the systematic modeling of operational transitions. The resulting MILP model is computationally very efficient, and therefore allows the solution of industrial-scale problems. However, in the proposed approach, a surrogate model is created for the plant as a whole, which does not explicitly consider interactions between different processes in the plant. Also, the model only considers day-ahead electricity prices and does not take purchasing electricity from pre-agreed power contracts into account.

In this work, we extend the model proposed by Mitra et al. (2012a) and further generalize it such that it can be applied to continuous process networks subject to various power contracts. In order to achieve this goal, we explicitly model the process network, in which processes are connected by material flows, and each process is represented by a CRS model. The mode-based scheduling formulation is applied, as well as a block contract formulation for the purpose of incorporating different types of power contracts.

3.2. Problem Statement

We consider process networks involving continuous processes which consume significant amount of electric energy during operation. Here, a process can refer to a piece of equipment, a set of multiple interconnected pieces of equipment, or an entire plant. The processes in such a network differ in their feeds and products, the restrictions on the production rates, and the power consumption characteristics. Inventory capacities are given for storable intermediate and final products. Demand is given for all final products.
Electricity can be purchased from various power contracts which may differ in price, availability, and penalty for under- or overconsumption. We assume that the prices and other conditions for negotiated contracts are known in advance, and that accurate forecasts for the day-ahead or real-time electricity prices are available.

The goal is to find a production schedule over a given time horizon that minimizes the total electricity cost while satisfying all product demand. For every time period of the scheduling horizon, we determine

- the mode of operation for each process,
- the processing rate in each process,
- the amounts of products stored,
- the amounts of products purchased,
- the amount of electricity purchased from each power contract.

### 3.3. Model Formulation

We propose an MILP scheduling model, for which the mathematical formulation is presented below. Note that all continuous variables in this model are constrained to be nonnegative.

#### 3.3.1. Time Representation

A discrete-time framework is applied in which the time horizon is divided into periods of equal length, $\Delta t$ (typically an hour). The notation for the time discretization is such that time period $t$ starts at time point $t-1$ and ends at time point $t$. The scheduling horizon is defined by the set of time periods $\mathcal{T} = \{1, 2, \ldots, \hat{t}\}$. Set $\overline{\mathcal{T}}$ is a subset of $\mathcal{T} = \{-\theta_{\text{max}} + 1, -\theta_{\text{max}} + 2, \ldots, 0, 1, \ldots, \hat{t}\}$, which also includes time periods in the past that are used in some constraints involving multiple time periods. The common-grid representation is illustrated in Figure 3.1.

#### 3.3.2. Process Network Representation

We consider a process network consisting of processes that are connected by material nodes, which represent potential storage units for feeds, intermediate products, and final products. The arcs in such a network depict the directions of the material
flows. Two or more streams entering the same material node are of the same quality, i.e. same chemical, temperature, pressure, etc., similarly as in the STN model (Kondili et al., 1993). Figure 3.2 shows an example of such a process network, in which the process and material nodes are depicted by rectangles and circles, respectively.

**Figure 3.2**: Simple process network consisting of process nodes (rectangles), material nodes (circles), and arcs depicting the material flows.

### 3.3.3. Mass Balance Constraints

For a given process network operating continuously in each time period $t$, the mass balance constraints can be stated as follows:

$$ Q_{jt} = Q_{j,t-1} + \sum_{i \in I_j} P_{ijt} - \sum_{i \in I_j} P_{jyt} + W_{jt} - D_{jt} \quad \forall j, \ t \in T $$  \hspace{1cm} (3.1a)

$$ Q^\text{min}_j \leq Q_{jt} \leq Q^\text{max}_j \quad \forall j, \ t \in T $$  \hspace{1cm} (3.1b)

$$ W_{jt} \leq W^\text{max}_j \quad \forall j, \ t \in T $$  \hspace{1cm} (3.1c)

where $Q_{jt}$ is the inventory level for material $j$ at time $t$, and $P_{ijt}$ is the amount of material $j$ consumed or produced by process $i$ in time period $t$. The set of processes
producing material $j$ is denoted by $\hat{I}_j$, whereas $\bar{I}_j$ is the set of processes receiving material $j$. The additional purchase of material $j$ in time period $t$ is denoted by $W_{jt}$. Parameter $D_{jt}$ denotes the demand for material $j$ in time period $t$. Eq. (3.1a) states that the inventory level of a material increases when it is produced or purchased, and the inventory level decreases when the material is consumed by other processes or used to meet product demand. Eq. (3.1b) sets lower and upper bounds on the inventory levels. For nonstorable materials, $Q_{j}^{\text{min}}$ and $Q_{j}^{\text{max}}$ are zero. Eq. (3.1c) limits the amount of material that can be purchased per time period.

3.3.4. Process Surrogate Model

In addition to the mass balance constraints that reflect the structure of the process network, we require a model for each individual process that represents its input-output relationship and expresses its capacity. Furthermore, for every feasible operating point, we have to know the amount of electricity required so that it can be included in the objective function. In other words, a description of the feasible operating region of each process is required in the space of its input and output materials, and an electricity consumption correlation that is a function of those inputs and outputs.

In this framework, it is assumed that each process can operate in different operating modes, where each mode represents a particular operating state, e.g. “off”, “on”, or “startup”. We propose to represent each mode with a CRS model; hence, the feasible operating region for a mode is defined by a union of convex subregions in the corresponding material space, and a linear electricity consumption function with respect to the production rates is given for each subregion. The key feature here is that every subregion has the form of a polytope. Figure 3.3 shows an illustrative example of the feasible operating region of a process in a two-dimensional material space. In this case, the materials are P1 and P2, and the process can operate in three different modes, Modes 1, 2, and 3. The feasible space of Mode 1 is a single point at the origin, which denotes zero production; thus, this mode is the “off” mode. The feasible space of Mode 2 is described by one single polytope, while Mode 3 is captured by the union of two polytopes, 3a and 3b. Note that the feasible regions of different modes may overlap as it is the case here for Modes 2 and 3.

Obviously, the feasible region of a real process typically does not have this polyhedral form. Usually, this is an approximation of the true feasible region. However, this representation allows us to formulate process models in a uniform manner, and
Figure 3.3: This illustrative example shows the feasible operating region of a process that can operate in three different operating modes.

As mentioned in Chapter 2, the resulting models are computationally efficient when embedded in an MILP scheduling model. For complex processes, CRS models can be constructed by using a model-based approach (Sung & Maravelias, 2009) or a data-driven algorithm as the one proposed in Chapter 2.

Physically, at any point in time, a process can only operate in one mode. For a given operating mode, the operating point has to lie in either one of the convex subregions. Any point in a subregion can be represented as a convex combination of the vertices of the polytope. These relationships can be expressed by the following nested disjunction:

\[
\bigvee_{m \in M_i} Y_{imt} \quad \forall i, t \in T
\]

\[
\bigvee_{r \in R_{im}} Y_{imt}
\begin{pmatrix}
Y_{imt} \\
Y_{imrt} \\
Y_{imrt} = \dfrac{P_{ijt}}{\phi_{imrj}} \quad \forall j \in J_i \\
\sum_{l \in L_{imr}} \lambda_{imrlt} = 1 \\
0 \leq \lambda_{imrlt} \leq 1 \quad \forall l \in L_{imr} \\
U_{it} = \delta_{imr} + \sum_{j \in J_i} \gamma_{imrj} P_{ijt}
\end{pmatrix}
\] \quad \forall i, t \in T \tag{3.2a}

\[
\bigvee_{m \in M_i} Y_{imt} \quad \forall i, t \in T
\] \quad \forall i, t \in T \tag{3.2b}

\[
Y_{imt} \Leftrightarrow \bigvee_{r \in R_{im}} Y_{imrt} \quad \forall i, m \in M_i, t \in T
\] \quad \forall i, m \in M_i, t \in T \tag{3.2c}

\[
Y_{imt} \in \{\text{true, false}\} \quad \forall i, m \in M_i, t \in T
\] \quad \forall i, m \in M_i, t \in T \tag{3.2d}
3. SCHEDULING OF PROCESS NETWORKS WITH POWER CONTRACTS

\[ Y_{imrt} \in \{ \text{true, false} \} \quad \forall \ i, \ m \in M_i, \ r \in R_{im}, \ t \in T \]  

(3.2e)

where \( M_i \) is the set of modes in which process \( i \) can operate, \( R_{im} \) is the set of operating subregions in mode \( m \in M_i \), \( L_{imr} \) is the set of vertices of subregion \( r \in R_{im} \), and \( J_i \) is the set of input and output materials of process \( i \). The Boolean variable \( Y_{imt} \) is true if mode \( m \in M_i \) is selected in time period \( t \), whereas the Boolean variable \( Y_{imrt} \) is true if subregion \( r \in R_{im} \) is selected in time period \( t \). The argument of the inner disjunction in (3.2a) states that the amount of material \( j \in J_i \) consumed or produced by process \( i \), \( P_{ijt} \), is expressed as a convex combination of the corresponding vertices, \( \phi_{imrt} \), while the amount of electricity consumed, \( U_{it} \), is a linear function of \( P_{ijt} \) with a constant \( \delta_{imr} \) and coefficients \( \gamma_{imrt} \) specific to the selected subregion. Eq. (3.2b) states that one and only one mode has to be selected for each process in each time period, and according to Eq. (3.2c), one region in \( R_{im} \) has to be selected if process \( i \) operates in mode \( m \in M_i \).

By applying the hull reformulation (Balas, 1985), the disjunction given by Eqs. (3.2) can be transformed into the following set of mixed-integer linear constraints:

\[ P_{ijt} = \sum_{m \in M_i} \sum_{r \in R_{im}} P_{imrt} \quad \forall \ i, j \in J_i, \ t \in T \]  

(3.3a)

\[ P_{imrt} = \sum_{l \in L_{imr}} \lambda_{imrt} \phi_{imrt} \quad \forall \ i, m \in M_i, r \in R_{im}, j \in J_i, t \in T \]  

(3.3b)

\[ \sum_{l \in L_{imr}} \lambda_{imrlt} = \bar{y}_{imt} \quad \forall \ i, m \in M_i, r \in R_{im}, t \in T \]  

(3.3c)

\[ U_{it} = \sum_{m \in M_i} \sum_{r \in R_{im}} \left( \delta_{imr} \bar{y}_{imt} + \sum_{j \in J_i} \gamma_{imrt} P_{imrt} \right) \quad \forall \ i, t \in T \]  

(3.3d)

\[ y_{imt} = \sum_{r \in R_{im}} \bar{y}_{imr} \quad \forall \ i, m \in M_i, t \in T \]  

(3.3e)

\[ \sum_{m \in M_i} y_{imt} = 1 \quad \forall \ i, t \in T \]  

(3.3f)

\[ y_{imt} \in \{0, 1\} \quad \forall \ i, m \in M_i, t \in T \]  

(3.3g)

\[ \bar{y}_{imrt} \in \{0, 1\} \quad \forall \ i, m \in M_i, r \in R_{im}, t \in T \]  

(3.3h)

where \( y_{imt} \) and \( \bar{y}_{imt} \) are binary variables, and \( P_{imrt} \) is the disaggregated variable. For each \( i, j \in J_i, t \in T \), only one \( P_{imrt} \) can be nonzero. Note that an equivalent formulation is achieved if \( y_{imt} \) is relaxed to be a continuous variable with \( 0 \leq y_{imt} \leq 1 \) since according to Eq. (3.3e), \( y_{imt} = \sum_{r \in R_{im}} \bar{y}_{imr} \), which restricts \( y_{imt} \) to be integer as \( \bar{y}_{imt} \) is binary.
3. SCHEDULING OF PROCESS NETWORKS WITH POWER CONTRACTS

3.3.5. Transition Constraints

A transition occurs when the system changes from one operating point to another. For changes between operating points belonging to the same operating mode, a bound on the rate of change, $\Delta_{imj}^{\text{max}}$, can be set with Eq. (3.4).

$$-\Delta_{imj}^{\text{max}} \leq \sum_{r \in R_{im}} \left( P_{imrjt} - P_{imrj,t-1} \right) \leq \Delta_{imj}^{\text{max}} \quad \forall \ i, m \in M_i, j \in J_i, t \in T$$ (3.4)

Additional constraints have to be imposed on transitions between different modes. The dynamics of a process in terms of its mode transition behavior can be visualized with a mode transition graph. Figure 3.4 shows an example with four different operating modes: off, on, startup, and shutdown. The arcs in the graph show the directions of the allowed transitions with the corresponding operational constraints. For instance, after shutting down the process, i.e. transitioning from the shutdown to the off mode, the process has to remain for at least 24 hours in the off mode before it can move to the startup mode. This constraint is typically imposed to reduce stress on the equipment. According to the mode transition graph, the startup phase takes exactly 4 hours before the process moves into its normal production mode, in which it has to remain for at least 48 hours.

![Mode transition graph showing the different operating modes and all possible transitions with the corresponding operational constraints.](image)

Eqs. (3.5)–(3.7), which model the transition constraints, are adopted from Mitra et al. (2012a, 2013). Here, $z_{imm't}$ is a binary variable which equals 1 if and only if process $i$ switches from mode $m$ to mode $m'$ at time $t$, which is enforced by the following constraint:

$$\sum_{m' \in TR_{im}} z_{imm't} - \sum_{m' \in \overline{TR}_{im}} z_{imm't} = y_{imt} - y_{im,t-1} \quad \forall \ i, m \in M_i, t \in T$$ (3.5a)

$$z_{imm't} \in \{0, 1\} \quad \forall \ i, (m, m') \in TR_i, t \in T$$ (3.5b)

where $\overline{TR}_{im} = \{m' : (m', m) \in TR_i\}$ and $\overline{TR}_{im} = \{m' : (m, m') \in TR_i\}$ with $TR_i$
being the set of all possible mode-to-mode transitions for process $i$.

The restriction that a process has to remain in a certain mode for a minimum amount of time after a transition is expressed by the following constraint:

$$y_{im'} \geq \sum_{k=1}^{\theta_{imm'}} z_{imm',t-k} \quad \forall i, (m, m') \in TR_i, t \in \mathcal{T}$$  \hspace{1cm} (3.6)

with $\theta_{imm'}$ being the minimum stay time in mode $m'$ after switching to it from mode $m$.

For predefined sequences, each of which defined as a fixed chain of transitions from mode $m$ to mode $m'$ to mode $m''$, we can specify a fixed stay time in mode $m'$ by imposing the following constraint:

$$z_{imm',t-\bar{\theta}_{imm'}m''} = z_{im'm''t} \quad \forall i, (m, m', m'') \in SQ_i, t \in \mathcal{T}$$  \hspace{1cm} (3.7)

where $SQ_i$ is the set of predefined sequences and $\bar{\theta}_{imm' m''}$ is the fixed stay time in mode $m'$ in the corresponding sequence. For instance, in the example shown in Figure 3.4, this constraint applies to the sequence off $\rightarrow$ startup $\rightarrow$ on. Since the startup process takes a certain amount of time, we can use Eq. (3.7) to fix the number of time periods in which the process has to remain in this mode once selected.

### 3.3.6. Energy Balance Constraints

The required amount of electricity can be purchased from multiple sources, which is stated in Eq. (3.8a). The different power sources are available power contracts, denoted by index $c$. Eq. (3.8b) sets lower and upper bounds on the electricity purchase from contract $c$ in time period $t$, $E_{ct}$. The lower bound, $E_{ct}^{\text{min}}$, is typically zero, but could be nonzero if the contract demands a minimum purchase.

$$\sum_t U_{it} = \sum_c E_{ct} \quad \forall t \in \mathcal{T}$$  \hspace{1cm} (3.8a)

$$E_{ct}^{\text{min}} \leq E_{ct} \leq E_{ct}^{\text{max}} \quad \forall c, t \in \mathcal{T}$$  \hspace{1cm} (3.8b)

### 3.3.7. Power Contract Model

Eqs. (3.8) are sufficient if power is only purchased from the spot market or from contracts that are merely defined by a unit price for each time period and possibly some minimum and maximum purchasing restrictions. However, large industrial electricity consumers typically commit themselves to power contracts that provide
additional favorable conditions. There are a large variety of such power contract structures; in this model, we consider the two most common types: discount and penalty contracts. With a discount contract, the unit price decreases with increasing amount of purchased electricity. For penalty contracts, the consumer agrees to either purchase at least a certain amount of electricity and pay a penalty for underconsumption, or not exceed a certain amount to avoid penalty for overconsumption.

Discount prices and penalties are defined with respect to the amount of electricity purchased over a certain period of time, which could be hours, days, or even weeks. In practice, this means that the cumulative electricity purchase is recorded, and there are predefined meter reading times at which the amount of electricity purchased since the last meter reading is computed. According to this cumulative electricity purchase between consecutive meter readings, discount prices and penalties are issued. In our model, we track the cumulative electricity purchase by using the following equations:

\[
F_{ct} = F_{c,t-1} + E_{ct} \quad \forall \ c \in \overline{C}, \ t \in \overline{T} \setminus \overline{T}_c \\
F_{ct} = F_{c,t-1} + E_{ct} - G_{ct} \quad \forall \ c \in \overline{C}, \ t \in \overline{T}_c \\
F_{ct} = 0 \quad \forall \ c \in \overline{C}, \ t \in \{0, \overline{T}_c\}
\]

where \(\overline{C}\) is the set of discount and penalty contracts and \(\overline{T}_c\) is the set of meter reading times for contract \(c\), which does not include time \(0\). The cumulative electricity purchase is denoted by \(F_{ct}\), which is reset to zero at every meter reading time. We further introduce the variable \(G_{ct}\), which is the cumulative electricity purchase since the last meter reading before time \(t\). To further clarify the notation, Figure 3.5 shows an illustrative example in which the cumulative electricity purchase meter readings are conducted every four time periods.

**Figure 3.5:** In this illustrative example, the cumulative electricity purchase meter is read every four hours.
To model discount and penalty contracts, we develop a block contract formulation that can accommodate both types of contracts. In a general block contract, each block is defined by an electricity price and the corresponding amount of electricity that one has to purchase in order to reach this block. In the case of a discount contract, the unit price decreases with each block as illustrated in Figure 3.6.

Figure 3.6: With a discount contract, unit price decreases with increasing electricity purchase. The ranges for the different prices define the contract blocks.

The following disjunction describes the mechanism of a block contract:

\[
\begin{align*}
\bigvee_{b \in B_c} & \quad X_{cbt} \\
& \quad \begin{cases} 
X_{cbt} & \quad H_{cbt} = H_{cb}^{\text{max}} \\
& \quad \forall b' \in B_c, b' < b \\
& \quad H_{cbt} \leq H_{cb}^{\text{max}} \\
& \quad H_{cbt} = 0 \quad \forall b' \in B_c, b' > b 
\end{cases} \\
\forall c \in \mathcal{C}, t \in \mathcal{\hat{T}}_c
\end{align*}
\]  

\[ (3.10a) \]

\[
\bigvee_{b \in B_c} X_{cbt} \quad \forall c \in \mathcal{C}, t \in \mathcal{\hat{T}}_c
\]

\[ (3.10b) \]

\[
X_{cbt} \in \{ \text{true, false} \} \quad \forall c \in \mathcal{C}, b \in B_c, t \in \mathcal{\hat{T}}_c
\]

\[ (3.10c) \]

where \( B_c \) is the set of blocks for contract \( c \), \( H_{cbt} \) denotes the amount of cumulative electricity purchased in block \( b \in B_c \) at time \( t \), and \( H_{cb}^{\text{max}} \) is the amount of electricity that one has to purchase in block \( b \in B_c \) before reaching the next block. The Boolean variable \( X_{cbt} \) is true if block \( b \) is the highest block reached for contract \( c \) at time \( t \). Disjunction (3.10a) states that if \( X_{cbt} \) is true, the maximum amount is purchased in all lower blocks \( b' < b \), the electricity purchase in block \( b \) is bounded by \( H_{cb}^{\text{max}} \), and no electricity is purchased in higher blocks \( b' > b \). According to logic constraint (3.10b), one and only one \( X_{cbt} \) has to be true.

Again, by applying the hull reformulation, Eqs. (3.10) can be transformed into
the following mixed-integer linear constraints:

\[
\sum_{b \in B_c} x_{cbt} = 1 \quad \forall \ c \in C, \ t \in \bar{T}_c \quad (3.11a)
\]

\[
\bar{H}_{cbt} = H_{cbt}^{\max} x_{cbt} \quad \forall \ c \in C, \ b \in B_c, \ b' \in B_c, \ b' < b, \ t \in \bar{T}_c \quad (3.11b)
\]

\[
\bar{H}_{cbt} \leq H_{cbt}^{\max} x_{cbt} \quad \forall \ c \in C, \ b \in B_c, \ t \in \bar{T}_c \quad (3.11c)
\]

\[
H_{cbt} = \sum_{b' \in B_c, b' < b} \Pi_{cb't} \quad \forall \ c \in C, \ b \in B_c, \ t \in \bar{T}_c \quad (3.11d)
\]

\[
x_{cbt} \in \{0, 1\} \quad \forall \ c \in C, \ b \in B_c, \ t \in \bar{T}_c \quad (3.11e)
\]

where \( x_{cbt} \) is a binary variable, and \( \bar{H}_{cbt} \) is the disaggregated variable.

To obtain the amount of electricity purchased from a contract, we sum up the electricity purchased in all corresponding contract blocks, as stated in Eq. (3.12).

\[
G_{ct} = \sum_{b \in B_c} H_{cbt} \quad \forall \ c \in C, \ t \in \bar{T}_c \quad (3.12)
\]

We can use the same block contract formulation to model penalty contracts by specifying \( H_{cb}^{\max} \) such that the first block (Block 1) and the last block (Block \( |B_c| \)) correspond to under- and overconsumption, respectively. A simple penalty contract is illustrated in Figure 3.7. Here, if \( H_{cbt} < H_{cb}^{\max} \) for \( b = 1 \), we underconsume, whereas if \( H_{cbt} > 0 \) for \( b = 3 \), we overconsume.

**Figure 3.7:** A simple penalty contract can be modeled as a three-block contract, with the first and third blocks corresponding to under- and overconsumption, respectively.
3.3.8. Boundary Conditions

We solve the scheduling problem for a given time horizon, with boundary conditions stated as follows:

\[
Q_{j,0} = Q_{j}^{ini} \quad \forall \ j \tag{3.13a}
\]
\[
y_{im,0} = y_{im}^{ini} \quad \forall \ i, m \in M_i \tag{3.13b}
\]
\[
z_{imm't} = z_{imm't}^{ini} \quad \forall \ i, (m, m') \in TR_i, -\theta_{i}^{max} + 1 \leq t \leq -1 \tag{3.13c}
\]
\[
Q_{j,i} \geq Q_{j}^{fin} \quad \forall \ j \tag{3.13d}
\]

with \(\theta_{i}^{max} = \max \left( \max_{(m,m') \in TR_i} \{\theta_{imm'}\}, \max_{(m,m',m'') \in SQ_i} \{\bar{\theta}_{imm'm''}\} \right)\), which defines for how far back in the past the mode switching information has to be provided. The initial conditions (3.13a)–(3.13c) set the initial inventory levels to \(Q_{j}^{ini}\), the initial operating modes to \(y_{im}^{ini}\), and the mode switching history according to \(z_{imm't}^{ini}\), while the terminal constraint Eq. (3.13d) sets the lower bounds on the final inventory levels to \(Q_{j}^{fin}\).

3.3.9. Objective Function

The objective is to minimize the total electricity cost, \(TC\), as expressed in the following objective function:

\[
TC = \sum_c \sum_{t \in T} \alpha_{ct} E_{ct} + \sum_{e \in C} \sum_{t \in T} \left[ \sum_{B_e} \beta_{ebt} H_{ebt} + \zeta_{cet}^{u}(H_{c,1}^{max} - H_{c,1,t}) + \zeta_{cet}^{o} H_{c,B_e,t} \right] \tag{3.14}
\]

where \(\alpha_{ct}\) and \(\beta_{cet}\) are unit costs for purchased electricity, whereas \(\zeta_{cet}^{u}\) and \(\zeta_{cet}^{o}\) are unit penalty costs for under- and overconsumption, respectively. The first term in Eq. (3.14) represents the base cost and applies to all contracts while the remaining terms only apply to block contracts.

Note that the price structure of a contract is defined by the cost coefficients \(\alpha_{ct}\), \(\beta_{cet}\), \(\zeta_{cet}^{u}\), \(\zeta_{cet}^{o}\), as well as \(H_{c,1}^{max}\). Many combinations are possible, which provides the flexibility of modeling various different power contracts. Some common examples are listed in the following:

- Day-ahead or real-time market: \(\alpha_{ct} > 0, \beta_{cet} = 0, \zeta_{cet}^{u} = 0, \zeta_{cet}^{o} = 0\)
- Pure discount contract: \(\alpha_{ct} = 0, \beta_{cet} > 0, \zeta_{cet}^{u} = 0, \zeta_{cet}^{o} = 0\)
- Contract with partial price discount: \(\alpha_{ct} > 0, \beta_{cet} > 0, \zeta_{cet}^{u} = 0, \zeta_{cet}^{o} = 0\)
3. SCHEDULING OF PROCESS NETWORKS WITH POWER CONTRACTS

- Contract with penalty for underconsumption: \( \alpha_{ct} > 0, \beta_{ct0} = 0, \zeta_{ctu} > 0, \zeta_{ct0} = 0 \)
- Contract with penalty for overconsumption: \( \alpha_{ct} > 0, \beta_{ct0} = 0, \zeta_{ctu} = 0, \zeta_{ct0} > 0 \)
- Contract with penalties for under- and overconsumption: \( \alpha_{ct} > 0, \beta_{ct0} = 0, \zeta_{ctu} > 0, \zeta_{ct0} > 0 \)
- Combined discount and penalty contract: \( \alpha_{ct} > 0, \beta_{ct0} > 0, \zeta_{ctu} > 0, \zeta_{ct0} > 0 \)

By considering the objective function given by Eq. (3.14), it is assumed that electricity cost constitutes the vast majority of the variable operating cost. If this assumption is invalid and additional costs—such as inventory costs, mode transition costs, and costs for purchasing material—need to be considered, they can be easily incorporated by assigning cost coefficients to the corresponding variables.

3.4. Numerical Results

In the following, we apply the proposed model to an illustrative example and a real-world industrial case study. All models were implemented in GAMS 24.4.1 (GAMS Development Corporation, 2015a), and the commercial solver CPLEX 12.6.1 (IBM ILOG, 2015a) was applied to solve the MILPs on an Intel® Core™ i7-2600 machine at 3.40 GHz with 8 processors and 8 GB RAM running Windows 7 Professional.

3.4.1. Illustrative Example

To demonstrate the main features of the model, we first apply it to an illustrative example for which the process network is shown in Figure 3.8.

Each process is characterized by its operating modes and operating subregions. The vertices of each convex subregion are listed in Table 3.1 where each vertex is given as a vector of the materials. Note that there is a different set of vertices for each subregion; however, in every vertex set, the numbering of the vertices starts at 1. Also, notice that the on mode of Process 3 is the only mode that is described by two subregions. Each operating subregion is further characterized by a linear electricity consumption function, which is shown in Table 3.2. Possible transitions between different operating modes and predefined sequences are listed in Tables 3.3 and 3.4, respectively; the tables also contain the corresponding minimum and fixed stay times given in hours. No bound is imposed on the rate of change in the same mode.
3. SCHEDULING OF PROCESS NETWORKS WITH POWER CONTRACTS

![Diagram of the process network](image)

**Figure 3.8:** The process network of the illustrative example consists of 4 process and 7 material nodes.

Data regarding the inventory of each material are given in Table 3.5. Infinite availability of feedstock, A and C, is assumed. Also note that Intermediate B is not storable. Moreover, no material can be purchased during the scheduling horizon, i.e. \( W_{j}^{\text{max}} = 0 \forall j \).

We consider a two-day scheduling horizon with an hourly time discretization. The results for three different product demand scenarios (low, high, and medium) will be compared; the corresponding demand profiles are shown in Figure 3.9. Note that in the low and high demand scenarios, the demands for E and G are constant over time while they vary in the medium demand scenario.

![Demand profiles](image)

**Figure 3.9:** Product demand profiles for the three different scenarios (low, high, and medium).

Three different power contracts are considered: a time-of-use (TOU), a discount, and a penalty contract. In addition to different base electricity price profiles, other features differentiating the three contracts are the following:

- The TOU contract does not have any additional price components.
3. SCHEDULING OF PROCESS NETWORKS WITH POWER CONTRACTS

Table 3.1: Vertices associated with each operating subregion of the processes from the illustrative example.

<table>
<thead>
<tr>
<th>Process</th>
<th>Mode</th>
<th>Region</th>
<th>Vertex</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>off</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>1</td>
<td>60</td>
<td>60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>600</td>
<td>600</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.2</td>
<td>off</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>200</td>
<td>200</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>startup</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>shutdown</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>1</td>
<td>40</td>
<td>20</td>
<td>60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>800</td>
<td>4000</td>
<td>1200</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>off</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>startup</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>shutdown</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>40</td>
<td>50</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>100</td>
<td>45</td>
<td>55</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>1000</td>
<td>500</td>
<td>500</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4</td>
<td>1000</td>
<td>450</td>
<td>550</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>100</td>
<td>45</td>
<td>55</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>100</td>
<td>40</td>
<td>60</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td>1000</td>
<td>450</td>
<td>550</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.1</td>
<td>off</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>300</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4.2</td>
<td>off</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>300</td>
<td>300</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- With the discount contract, the additional cost depends on the purchased amount of electricity over the course of each day, i.e. the cumulative electricity consumption meter is read at the end of each day. For the first 50 MWh, an additional cost of $10/MWh needs to be paid; for the next 40 MWh, a cost of $8/MWh occurs; and for any amount of electricity purchased beyond 90 MWh, the cost is $5/MWh.
- With the penalty contract, penalties are applied to under- as well as over-consumption. A penalty of $50/MWh has to be paid if the daily electricity purchase from the penalty contract is below 20 MWh or exceeds 80 MWh.
3. SCHEDULING OF PROCESS NETWORKS WITH POWER CONTRACTS

Table 3.2: Electricity consumption correlations associated with each operating sub-region. Each correlation is a linear function of the materials.

<table>
<thead>
<tr>
<th>Process</th>
<th>Mode</th>
<th>Region</th>
<th>Electricity Consumption Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>off</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>500 + 2A</td>
</tr>
<tr>
<td>1.2</td>
<td>off</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>450 + 3A</td>
</tr>
<tr>
<td>2</td>
<td>off</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>startup</td>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>shutdown</td>
<td>1</td>
<td>150</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>100 + 0.5D</td>
</tr>
<tr>
<td>3</td>
<td>off</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>startup</td>
<td>1</td>
<td>400</td>
</tr>
<tr>
<td></td>
<td>shutdown</td>
<td>1</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>800 + 3D</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>2</td>
<td>1000 + 4D</td>
</tr>
<tr>
<td>4.1</td>
<td>off</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>400 + 2F</td>
</tr>
<tr>
<td>4.2</td>
<td>off</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on</td>
<td>1</td>
<td>350 + 1.5F</td>
</tr>
</tbody>
</table>

Table 3.3: Possible transitions between the different operating modes of each process and the corresponding minimum stay times.

<table>
<thead>
<tr>
<th>Process</th>
<th>Transition from Mode m to Mode m’</th>
<th>Minimum Stay Time in Mode m’</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>off → on</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on → off</td>
<td>0</td>
</tr>
<tr>
<td>1.2</td>
<td>off → on</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on → off</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>off → startup</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>startup → on</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>on → shutdown</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>shutdown → off</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>off → startup</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>startup → on</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>on → shutdown</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>shutdown → off</td>
<td>12</td>
</tr>
<tr>
<td>4.1</td>
<td>off → on</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on → off</td>
<td>0</td>
</tr>
<tr>
<td>4.2</td>
<td>off → on</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>on → off</td>
<td>0</td>
</tr>
</tbody>
</table>

Otherwise, there is no cost in addition to the base cost.

Moreover, when applied, the maximum amount that can be purchased from each
Table 3.4: Predefined sequences of mode transitions and the corresponding fixed stay times.

<table>
<thead>
<tr>
<th>Process</th>
<th>Sequence (Transition from ( m ) to ( m' ) to ( m'' ))</th>
<th>Fixed Stay Time in ( m' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>off → startup → on</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>on → shutdown → off</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>off → startup → on</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>on → shutdown → off</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 3.5: Inventory bounds and initial inventory levels for each material.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum Inventory</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3000</td>
<td>0</td>
<td>5000</td>
<td></td>
</tr>
<tr>
<td>Maximum Inventory</td>
<td>( \infty )</td>
<td>( \infty )</td>
<td>8000</td>
<td>20,000</td>
<td>2000</td>
<td>30,000</td>
<td></td>
</tr>
<tr>
<td>Initial Inventory</td>
<td>( \infty )</td>
<td>( \infty )</td>
<td>800</td>
<td>5000</td>
<td>0</td>
<td>8000</td>
<td></td>
</tr>
<tr>
<td>Minimum Final Inventory</td>
<td>0</td>
<td>0</td>
<td>800</td>
<td>5000</td>
<td>0</td>
<td>8000</td>
<td></td>
</tr>
</tbody>
</table>

contract every hour is 10 MWh.

Using the data from the illustrative example, we now consider five cases which differ in the product demand and the applied power contracts as shown in Table 3.6. In all cases, all processes are in the on mode at the start of the scheduling horizon. Also, it is assumed that no mode switching has occurred in the previous 12 hours. The total electricity costs in the five cases obtained by solving the MILP model are listed in the last column of Table 3.6.

Table 3.6: The table lists the five cases, which differ in the product demands and in the power contracts, with the optimal total electricity costs.

<table>
<thead>
<tr>
<th>Case</th>
<th>Product Demand</th>
<th>Power Contracts</th>
<th>Total Electricity Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>low</td>
<td>TOU</td>
<td>$5771</td>
</tr>
<tr>
<td>2</td>
<td>high</td>
<td>TOU</td>
<td>$14,624</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>TOU</td>
<td>$11,697</td>
</tr>
<tr>
<td>4</td>
<td>medium</td>
<td>TOU, discount</td>
<td>$11,057</td>
</tr>
<tr>
<td>5</td>
<td>medium</td>
<td>TOU, penalty</td>
<td>$11,346</td>
</tr>
</tbody>
</table>

For Case 1, the optimal schedule is shown in Figure 3.10, which depicts a Gantt chart showing the operating modes of each process in each time period. Figure 3.11 shows the amount of electricity consumed by each process. Also, one can observe that the plant is shut down for a large portion of the time, primarily when the electricity price is high. This large shift in load is possible due to the low demand and the flexibility in the inventory as indicated by the changes in inventory levels shown in Figure 3.12. Also note that for the conversion of A to B, Process
1.1 is preferred over Process 1.2 due to its higher capacity and lower unit electricity consumption. Similarly, Process 4.2 is preferred over Process 4.1 due to its higher efficiency.

![Gantt chart for the optimal schedule in Case 1. Selected operating modes are shown for each time period.](image)

**Figure 3.10:** Gantt chart for the optimal schedule in Case 1. Selected operating modes are shown for each time period.

![Electricity consumption profiles in Case 1](image)

**Figure 3.11:** Amount of electricity consumed by each process in Case 1.

The electricity consumption profiles in Case 2 are shown in Figure 3.13, which indicates only moderate load shifting. This can be explained by the reduced level of process flexibility due to the high demand. Here, the plant has to utilize almost its entire production capacity to satisfy demand. Figure 3.14 shows the result for Case 3, which indicates that with medium time-varying demand, there is again significant potential for load shifting.

In Case 4, the discount contract is applied in addition to the TOU contract. Figure 3.15 shows the breakdown of the total electricity purchase into the purchases from these two contracts and the corresponding base prices. The majority of the consumed electricity in the first 24 hours, namely 135.5 MWh, is purchased from the discount contract. By doing so, we take advantage of the best possible discount. However, there are still time periods in which the TOU contract offers a lower price. In those hours, electricity is purchased from the TOU contract. On the second day,
3. SCHEDULING OF PROCESS NETWORKS WITH POWER CONTRACTS

Figure 3.12: Inventory profiles of the intermediate materials D and F, and the final products E and G in Case 1.

Figure 3.13: Amount of electricity consumed by each process in Case 2.

53.9 MWh are purchased from the discount contract. We would purchase more from the discount contract for a larger discount; however, the threshold for that discount is too high for this to be beneficial. Therefore, the rest of the consumed electricity is purchased from the TOU contract.

In Case 5, the TOU and the penalty contracts are applied. Figure 3.16 shows the breakdown of the total electricity purchase and the base prices. Notice that the price for the penalty contract is higher on the second day. 80 MWh and 20 MWh are purchased from the penalty contract on the first and second day, respectively. It is clear that because of the low price on the first day, we try to purchase as much as possible from the penalty contract without having to pay any penalties. On the second day, since the price is now high, we purchase just enough to avoid penalties.
3. SCHEDULING OF PROCESS NETWORKS WITH POWER CONTRACTS

Figure 3.14: Amount of electricity consumed by each process in Case 3.

Figure 3.15: Breakdown of the total electricity purchase into the purchases from the TOU and the discount contracts in Case 4.

Figure 3.16: Breakdown of the total electricity purchase into the purchases from the TOU and the penalty contracts in Case 5.

In each case, the model has approximately 5000 continuous variables, 2500 binary variables, and 7500 constraints. All models were solved to zero integrality gap in less than 20 seconds wall-clock time.
3.4.2. *Industrial Case Study*

We now apply the proposed model to a real-world industrial case study provided by Praxair. Here, we consider an air separation plant that produces gaseous oxygen (GO2), gaseous nitrogen (GN2), liquid oxygen (LO2), liquid nitrogen (LN2), and liquid argon (LAr). The corresponding process network is shown in Figure 3.17. While GO2, LO2, LN2, and LAr can be directly sold to the market, GN2 has to be further compressed before it can be supplied to the customers. Two kinds of GN2 are sold: medium-pressure GN2 (MPGN2) and high-pressure GN2 (HPGN2). GN2 is compressed to MPGN2 through Process LMCompGN2 and can be further compressed to HPGN2 through Process MHCompGN2; it can also be directly converted to HPGN2 by running Process LHCompGN2. Furthermore, GN2 can be liquefied to LN2 through Process LiqGN2. Overproduced gaseous products can be vented through a venting process, and all liquid products can be converted into the corresponding gaseous products through a so-called driox process. The CRS models for the processes have been generated by applying the data-driven algorithm proposed in Chapter 1.

![Figure 3.17: Process network representing the given air separation plant.](image_url)

While there are inventories of the liquid products, we assume that gaseous products cannot be stored. Electricity can be purchased from three different sources: a TOU contract, a penalty contract, and the spot market. While pre-agreed prices for the TOU and the penalty contracts are known, a forecast of the hourly-
varying spot price is used in the model. For the penalty contract, penalties are only paid for underconsumption.

The scheduling horizon is one week, to which an hourly time discretization is applied resulting in 168 time periods. Demand profiles for all products are given. Note that while there is continuous demand for gaseous products, the demand for each liquid product is assumed to occur only at the end of each day. The assumption here is that there is sufficient capacity in the inventory to handle flows into and out of the inventory tank throughout the day. The demand at the end of the day is then the total amount of product that needs to be drawn from the tank over the course of the day.

We solve the resulting MILP model to obtain the optimal schedule. Figure 3.18 shows the electricity consumption profiles for each of the processes. The vast majority of the electricity consumption is attributed to the air separation unit (ASU). The GN2 liquefier also consumes a large amount of electricity but is only used five times, each time for a few hours. Compared to the ASU and the GN2 liquefier, the pipeline compressors contribute relatively little to the total electricity consumption. Significant load shifting can be observed in the schedule; this is mainly realized by operating the liquefier during low-price hours, which allows a fairly constant operation of the other processes.

![Figure 3.18: Amount of electricity consumed by each process of the air separation plant.](image)

The electricity prices for the TOU and the penalty contracts as well as the spot electricity price are shown in Figure 3.19. The breakdown of the total electricity purchase into the purchases from the three different sources is shown in Figure 3.20. One can observe that in each time period, we choose to purchase from the source with the lowest price. Two sources are chosen in the same time period only when the maximum purchase amount is reached for one of the two sources. Also,
sufficient amount of electricity is purchased from the penalty contract such that no penalty has to be paid.

![Graph showing electricity prices for TOU, penalty, and spot contracts](image)

**Figure 3.19:** Electricity prices for the TOU contract, the penalty contract, and the spot market.

![Bar chart showing breakdown of electricity purchase](image)

**Figure 3.20:** Breakdown of the total electricity purchase into the purchases from the three different sources.

As an example, Figure 3.21 shows the amount of GN2 produced by the AS process and the GN2 flows into the subsequent processes. Large portion of GN2 is compressed to feed the product pipelines. Another significant amount of GN2 is liquefied to increase the production of LN2. However, the majority of GN2 is vented. The reason for this overproduction of GN2 is that the plant is “oxygen-limited”, which means that the production is driven by the oxygen demand. At such a plant, the nitrogen production usually exceeds the demand, which makes venting of GN2 necessary.

Figure 3.22 shows the amounts of LN2 produced by the ASU and the GN2 liquefier in each time period. The inventory increases when LN2 is produced and it decreases when LN2 is drawn from the tank to satisfy demand, which occurs every
3. SCHEDULING OF PROCESS NETWORKS WITH POWER CONTRACTS

Figure 3.21: GN2 production and its breakdown into feeds for different processes.

24 hours and is not shown in the figure. One can see that in addition to the pro-
duction through the AS process, a large amount of LN2 is obtained by liquefying
GN2. In this solution, driox is not used in any of the time periods.

Figure 3.22: LN2 production and inventory profile.

The MILP model for the industrial-scale problem has 28,808 continuous vari-
ables, 9,776 binary variables, and 113,810 constraints. It was solved to zero inte-
grality gap in 7 seconds wall-clock time. A computational study was conducted by
solving several instances of the problem with different input parameters. Almost
all instances were solved within two minutes.

3.5. Summary

In this chapter, a general discrete-time MILP model has been developed for the
scheduling of continuous power-intensive process networks with various power
contracts. The main focus of the proposed formulation are the accurate and ef-
ficient representation of the operational flexibility in the production process and
the modeling of power contracts. A process network has been considered where
each process is represented by a CRS model. A mode-based formulation has been
adopted to model operational transitions. Moreover, a block contract formulation has been proposed that allows the modeling of a large variety of power contracts including commonly occurring discount and penalty contracts.

The proposed scheduling model has been applied to an illustrative example as well as to a real-world air separation plant. The results demonstrate the capability of the model in representing the operation of continuous process networks and the flexibility in modeling different power contracts. This allows the optimal scheduling of power-intensive processes involving load shifting within the feasible range of operation. Furthermore, the proposed MILP model has proven to be computationally very efficient. Large-scale problems with tens of thousands of variables and hundreds of thousands of constraints can be solved within a few minutes, which allows the use of such a scheduling tool in a real industrial setting. In fact, a scheduling tool using the proposed modeling framework is currently implemented in one of Praxair’s air separation plants.
4. **RISK-BASED INTEGRATED PRODUCTION SCHEDULING AND ELECTRICITY PROCUREMENT**

In the previous chapter, we developed a deterministic scheduling model that optimizes production scheduling and electricity procurement decisions simultaneously. Now, in this chapter, we take the next step in which we also consider uncertainty. For this purpose, a stochastic programming model is developed, which accounts for the two most common sources of uncertainty in industrial DSM: spot electricity price and product demand. Conditional value-at-risk is incorporated into the model as a measure of risk. A main focus of this work is the comparison between deterministic, stochastic risk-neutral, and stochastic risk-averse optimization. The results show that the solutions obtained from these three approaches can differ very significantly.

This chapter is organized as follows. Section 4.1 provides a literature review on electricity procurement scheduling and its integration with production scheduling. The problem statement is presented in Section 4.2 before the MILP model is developed in Section 4.3. Section 4.4 describes the applied scenario generation and reduction techniques, while Section 4.5 outlines the multicut Benders decomposition algorithm used to solve large-scale instances. In Section 4.6, the proposed model is applied to an illustrative example and an industrial case study. In Section 4.7, we close with a summary of the results.

4.1. **Background**

Due to high fluctuations in electricity demand and increased integration of intermittent renewable energy, electricity prices have become highly volatile and difficult to predict, which poses immense challenges to power-intensive industries. For large industrial electricity consumers, there are two ways of dealing with uncertainty in electricity price: (1) dynamically adjust the production schedule to changes in the spot price, i.e. shift the electricity load to lower-price periods; (2) re-
move price uncertainty by signing power contracts with agreed fixed prices. Both strategies can be very effective in reducing the electricity cost, but they also have their limitations and drawbacks. A plant’s capability for demand response is limited by the flexibility of the production process, which has to be carefully evaluated in order to avoid detrimental disruptions caused by sudden changes in the plant operation. Power contracts provide fixed electricity prices; however, this reduction in risk usually comes at the cost of higher expected average prices. Moreover, power contracts require the consumers to commit themselves in advance to the amount that they are going to purchase for a certain period of time. This commitment reduces the consumers’ demand response opportunities since there is less room for adjustments in response to real-time price changes. Hence, there is a trade-off between purchasing power from contracts and from the spot market.

It is clear that often only a combination of the two aforementioned strategies will lead to the best result. Here, the major challenge in the decision-making is uncertainty. This uncertainty does not only occur in the electricity price; another source of uncertainty that has a possibly even greater impact on the production schedule is product demand. Major operational decisions and decisions regarding the commitment to power contracts have to be made before the actual spot electricity price and product demand are known for the time horizon of interest. There is only limited room for reactive actions as soon as these decisions are made. Therefore, it is crucial to account for these uncertainties in the decision-making process.

Although many works have addressed problems involving power contracts from an electricity producer’s or retailer’s point of view (Conejo et al., 2008; Lima et al., 2015; Carrión et al., 2007; Hatami et al., 2009b), the literature is scarce in papers considering the consumer’s perspective. Conejo et al. (2005) solve a medium-term electricity procurement problem that considers a set of bilateral contracts, hourly changing spot prices, and the possibility of producing electricity with an onsite power generation facility. The self-generated power can be used for own consumption or sold to the spot market. A subsequent work (Conejo & Carrión, 2006) addresses a similar problem for a shorter time horizon, while considering cost volatility by using an estimate of the covariance of the spot price. While the models proposed in these two papers are deterministic, Carrión et al. (2007) apply stochastic programming to explicitly model uncertainty in electricity prices; furthermore, the conditional value-at-risk (CVaR) is included in the model as a measure of risk, which is used to show the clear trade-off between expected cost and risk. A similar trade-off is shown by Zare et al. (2010) who apply the concept of information gap.
decision theory to evaluate the robustness of a solution against high spot prices or high procurement costs. Beraldi et al. (2011) consider the short-term electricity procurement problem involving bilateral contracts and the day-ahead market; here, a stochastic programming model is solved in a rolling-horizon fashion.

In all the works reviewed in the previous paragraph, the consumer’s electricity demand profile is assumed to be known and therefore fixed. This implies that a separate production scheduling problem has to be solved first in order to determine the electricity demand, which then can be used as input in the electricity procurement problem. However, this sequential approach is likely to be suboptimal since the production scheduling problem does not take the full electricity price information into account.

In the work presented in this chapter, in addition to taking an integrated approach, we apply stochastic programming to model uncertainty in both spot electricity price and product demand, which to the best of our knowledge has not been considered in this context before. Also, while most stochastic programming models in the literature only optimize the expected outcome, we acknowledge the importance of accounting for risk in the decision-making process, which we accomplish by incorporating the CVaR into the proposed model. Furthermore, scenario reduction and multicut Benders decomposition are applied to the resulting MILP model in order to solve large-scale industrial problems.

4.2. Problem Statement

We consider a power-intensive continuously operated plant that can produce a given set of products. Inventory capacities exist for storable products, and additional products can be purchased at given costs. It is assumed that for fixed product demand, all production costs, besides the cost of electricity, are constant. In this way, for optimization purposes, the total operating cost only consists of the electricity cost and the cost of purchasing additional products.

Electricity can be purchased from the spot market or from power contracts that have fixed pre-agreed electricity price and availability conditions. While purchases from the spot market can be made a day in advance (day-ahead) or on the spot (real-time), one has to commit to the electricity purchase from power contracts for a longer period of time, e.g. for one week.

The goal is to optimize the production and electricity procurement schedules in terms of expected cost and/or risk over a given time horizon. For this purpose, un-
certainty in spot electricity price and product demand is considered. The decisions can be divided into two sets: one containing here-and-now decisions that have to be made at the beginning and cannot be changed over the course of the scheduling horizon, the other containing wait-and-see decisions that can be adjusted after realization of the uncertainty. In this problem, the here-and-now decisions are the mode of operation for the production process and the amount of electricity purchased from each power contract for each time period of the scheduling horizon. The wait-and-see decisions are the actual production rates, the amounts of products stored, the amounts of products purchased, and the amount of electricity purchased from the spot market.

4.3. Model Formulation

The stochastic scheduling problem is formulated as an MILP. The underlying production scheduling model is a special case of the model presented in Chapter 3. Here, the plant is modeled as a single process rather than a process network, and the feasible region of each operating mode takes the form of a single polytope rather than the union of multiple polytopes. The full stochastic model is presented in the following. Unless specified otherwise, all continuous variables in this model are constrained to be nonnegative. Also, note that the nomenclature applied in this chapter differs from the one in Chapter 3.

4.3.1. Uncertainty Modeling Strategy

We adopt a stochastic programming (Birge & Louveaux, 2011) approach to model the uncertainty in electricity price and product demand. In stochastic programming, uncertainty is represented by discrete scenarios, and decisions are made at different stages, which are defined such that realization of uncertainty is observed between two stages, and at each stage, actions depending on previous observations are taken.

Depending on the type of spot market, the real electricity price can be observed minutes (real-time), hours, or one day (day-ahead) in advance. Also, production rates can be adjusted in every time period, which makes this problem a multistage problem. However, the resulting multistage stochastic programming problem is extremely large and computationally intractable. Therefore, we approximate the multistage problem with a two-stage stochastic programming problem where we assume that all uncertainty for the entire scheduling horizon is realized right after
the here-and-now decisions are made.

We define the set of product demand scenarios, \( S^D \), with probabilities \( \varphi^D_s \) of each demand scenario \( s \); similarly, \( S^P \) denotes the set of electricity price scenarios with \( \varphi^P_s \) being the probability of price scenario \( s \). Each pair of demand scenario \( s \) and price scenario \( \hat{s} \) corresponds to a general scenario \( s \in S \) with the probability \( \varphi_s = \varphi^D_s \varphi^P_s \). For the sake of brevity, we refer to the general scenario set \( S \) in most part of the model formulation.

4.3.2. Plant Model

In this framework, we assume that the plant can operate in different operating modes. For each mode, the feasible operating region is defined by a polytope in the product space, and a linear electricity consumption function with respect to the production rates is given for each mode. At any point in time, the plant can only run in one operating mode. For a given operating mode, the operating point has to lie within the corresponding polytope. These relationships can be described by the following constraints:

\[
PD_{its} = \sum_m PD_{mits} \quad \forall \; i, t \in T, s \quad (4.1a)
\]

\[
PD_{mits} = \sum_{j \in J_m} \lambda_{mjt} v_{mji} \quad \forall \; m, i, t \in T, s \quad (4.1b)
\]

\[
\sum_{j \in J_m} \lambda_{mjt} = y_{mt} \quad \forall \; m, t \in T, s \quad (4.1c)
\]

\[
EU_{ts} = \delta_m y_{mt} + \sum_i \gamma_{mi} PD_{mits} \quad \forall \; t \in T, s \quad (4.1d)
\]

\[
\sum_m y_{mt} = 1 \quad \forall \; t \in T \quad (4.1e)
\]

\[
y_{mt} \in \{0, 1\} \quad \forall \; m, t \in T \quad (4.1f)
\]

where \( J_m \) is the set of vertices of the polytope associated with mode \( m \). The binary variable \( y_{mt} \) equals 1 if mode \( m \) is selected in time period \( t \). The amount of product \( i \) produced in time period \( t \) of scenario \( s \) is denoted by \( PD_{its} \). Associated with \( PD_{its} \) is the disaggregated variable \( PD_{mits} \) for mode \( m \), which is expressed as a convex combination of the corresponding vertices, \( v_{mji} \). The amount of electricity consumed, \( EU_{ts} \), is a linear function of \( PD_{its} \) with a constant \( \delta_m \) and coefficients \( \gamma_{mi} \) specific to the selected mode. Note that while \( y_{mt} \) is a first-stage variable, \( PD_{its} \) and \( EU_{ts} \) are second-stage variables that depend on scenario \( s \).
4.3.3. Transition Constraints

A transition occurs when the system changes from one operating point to another. In particular, constraints have to be imposed on transitions between different operating modes, which is achieved by Eqs. (4.2)–(4.4). The binary variable $z_{mm't}$ equals 1 if and only if the plant switches from mode $m$ to mode $m'$ at time $t$, which is enforced by the following constraint:

$$\sum_{m' \in \overline{TR}_m} z_{m'm,t-1} - \sum_{m' \in \overline{TR}_m} z_{mm',t-1} = y_{mt} - y_{m,t-1} \quad \forall \ m, \ t \in \overline{T}$$  \hspace{1cm} (4.2a)

$$z_{mm't} \in \{0, 1\} \quad \forall \ (m, m', t) \in TR, \ t \in \overline{T}$$  \hspace{1cm} (4.2b)

where $\overline{TR}_m = \{m' : (m', m) \in TR\}$ and $\overline{TR}_m = \{m' : (m, m') \in TR\}$ with $TR$ being the set of all possible mode-to-mode transitions.

The restriction that the plant has to remain in a certain mode for a minimum amount of time after a transition is expressed in the following constraint:

$$y_{m't} \geq \overline{\theta}_{mm'} \sum_{k=1}^{\overline{\theta}_{mm'}} z_{mm',t-k} \quad \forall \ (m, m') \in TR, \ t \in \overline{T}$$  \hspace{1cm} (4.3)

with $\theta_{mm'}$ being the minimum stay time in mode $m'$ after switching to it from mode $m$.

For predefined sequences, each defined as a fixed chain of transitions from mode $m$ to mode $m'$ to mode $m''$, we can specify a fixed stay time in mode $m'$ by imposing the following constraint:

$$z_{mm',t-\overline{\theta}_{mm'm''}} = z_{m'm,t} \quad \forall \ (m, m', m'') \in SQ, \ t \in \overline{T}$$  \hspace{1cm} (4.4)

where $SQ$ is the set of predefined sequences and $\overline{\theta}_{mm'm''}$ is the fixed stay time in mode $m'$ in the corresponding sequence.

4.3.4. Mass Balance Constraints

The plant produces a certain set of products, of which some may be storable. As stated in Eq. (4.5a), the inventory level at time $t$, $IV_{its}$, is the inventory level at time $t - 1$ plus the amount produced minus the amount sold, $SL_{its}$, and minus the amount wasted, $PW_{its}$, in time period $t$. $PW_{its}$ takes a nonzero value if the demand is satisfied and the inventory has reached its maximum capacity. Eq. (4.5b) sets bounds on the inventory levels, and Eq. (4.5c) states that also products purchased
from other sources, denoted by $PC_{its}$, can be used to satisfy demand. Note that all variables involved in the mass balance constraints are second-stage variables.

$$ IV_{its} = IV_{i,t-1,s} + PD_{its} - SL_{its} - PW_{its} \quad \forall \ i, t \in T, s $$

$$ IV_{it}^{\text{min}} \leq IV_{its} \leq IV_{it}^{\text{max}} \quad \forall \ i, t \in T, s $$

$$ SL_{its} + PC_{its} = D_{its} \quad \forall \ i, t \in T, s $$

### 4.3.5. Energy Balance Constraints

As stated in Eq. (4.6a), the plant can be powered by electricity purchased from contracts and from the spot market, denoted by $EC_{ct}$ and $ES_{ts}$, respectively. $EW_{ts}$ is the amount of electricity “wasted” in time period $t$ of scenario $s$, which takes a nonzero value if the committed electricity purchase from contracts exceeds the electricity consumption in that particular scenario $s$. In practice, this amount of “wasted” electricity is not delivered, yet you still have to pay for it. Here, $EC_{ct}$ is a first-stage variable, whereas $ES_{ts}$ and $EW_{ts}$ are second-stage variables. Eq. (4.6b) restricts the amount of electricity that can be purchased from the spot market in each time period.

$$ EU_{ts} = \sum_c EC_{ct} + ES_{ts} - EW_{ts} \quad \forall \ t \in T, s $$

$$ ES_{ts} \leq ES_{t}^{\text{max}} \quad \forall \ t \in T, s $$

### 4.3.6. Power Contract Model

Power contracts can be very complex in their price structures. Here, we apply a relatively simple model that incorporates the main features of any common power contract. We assume that for each contract $c$, the price consists of two components: a time-dependent, and an amount-dependent component. The time-dependent price component roughly follows the expected spot electricity price profile, and is typically given for so-called time-of-use (TOU) periods. Figure 4.1a shows an example with four TOU periods over the course of 24 hours. The time-dependent price level depends on the type of TOU period: off-peak, mid-peak, or on-peak. The amount-dependent component sets a base price that depends on the total amount of electricity purchased during the entire scheduling horizon. As illustrated in Figure 4.1b, electricity is offered at discounted rates when certain purchase amounts have been reached.
4. Risk-based Production Scheduling and Electricity Procurement

Figure 4.1: Contract prices typically consist of a time-dependent and an amount-dependent component.

The time-dependent price component is simply expressed through the cost coefficient $\alpha_{ct}$ associated with the variable $EC_{ct}$, which is the amount of electricity purchased from contract $c$ in time period $t$. To accommodate the amount-dependent price component, we apply a block contract model which can be formulated as follows:

$$\overline{EC}_c = \sum_{t \in T} EC_{ct} \quad \forall \ c$$  \hspace{1cm} (4.7a)

$$\bigvee_{b \in B_c} \left[ X_{cb} \right]$$

$$BC_c = \beta_{cb} \overline{EC}_c$$

$$\overline{EC}_{c,b-1} \leq \overline{EC}_c \leq \overline{EC}_{cb}$$

$$\forall \ c$$  \hspace{1cm} (4.7b)

$$\bigvee_{b \in B_c} X_{cb} \quad \forall \ c$$  \hspace{1cm} (4.7c)

$$X_{cb} \in \{\text{true, false}\} \quad \forall \ c, b \in B_c$$  \hspace{1cm} (4.7d)

where $\overline{EC}_c$ is the amount of electricity purchased from contract $c$ over the entire scheduling horizon. The set of blocks of contract $c$ is denoted by $B_c$, $X_{cb}$ is a Boolean variable that is true if block $b$ of contract $c$ is chosen, $\overline{EC}_{cb}$ is the amount of electricity that one has to purchase in block $b \in B_c$ before reaching the next block. Hence, block $b$ is chosen if $\overline{EC}_c$ takes a value between $\overline{EC}_{c,b-1}$ and $\overline{EC}_{cb}$. In that case, the base cost, $BC_c$, is $\overline{EC}_c$ times the corresponding unit cost coefficient $\beta_{cb}$. Eq. (4.7c) states that only one block can be chosen for each contract.

By applying the hull reformulation (Balas, 1985), the disjunction and the logic constraints given by Eqs. (4.7b)–(4.7d) can be transformed into the following set of
mixed-integer linear constraints:

\[
\bar{E}_c = \sum_{b \in B_c} \bar{E}_{cb} \quad \forall c
\]  
(4.8a)

\[
BC_c = \sum_{b \in B_c} \beta_{cb} \bar{E}_{cb} \quad \forall c
\]  
(4.8b)

\[
\bar{E}_{c,b-1} x_{cb} \leq \bar{E}_{cb} \quad \forall c, b \in B_c, b > 1
\]  
(4.8c)

\[
\bar{E}_{cb} \leq \bar{E}_{c,b}^{\text{max}} x_{cb} \quad \forall c, b \in B_c
\]  
(4.8d)

\[
\sum_{b \in B_c} x_{cb} = 1 \quad \forall c
\]  
(4.8e)

\[
x_{cb} \in \{0, 1\} \quad \forall c, b \in B_c
\]  
(4.8f)

where \(x_{cb}\) is a binary variable, and \(\bar{E}_{cb}\) is the disaggregated variable associated with \(\bar{E}_c\).

As stated before, we assume that contract terms require that the electricity purchase decisions for the scheduling horizon are made before the beginning of that time horizon. These decisions cannot be changed later, i.e. consumers have to purchase the amount of electricity to which they have committed themselves regardless of their actual need. Hence, all contract-related decisions are first-stage decisions. Moreover, in each TOU period, which generally consists of multiple time periods, the power purchase from a contract has to remain constant for the entire duration of that TOU period, which is stated in the following constraint:

\[
EC_{ct} = EC_{cp} \quad \forall c, p \in P_c, t \in \hat{T}_{cp}
\]  
(4.9a)

\[
EC_{cp} \leq EC_{c,p}^{\text{max}} \quad \forall c, p \in P_c
\]  
(4.9b)

where \(EC_{cp}\) is the electricity purchased from contract \(c\) in each time period within the TOU period \(p\). The set of TOU periods for contract \(c\) is denoted by \(P_c\), while \(\hat{T}_{cp}\) is the set of time periods in TOU period \(p\) of contract \(c\). The amount that can be purchased from contract \(c\) in TOU period \(p\) is bounded by \(EC_{cp}^{\text{max}}\).

### 4.3.7. Boundary Conditions

The initial conditions and terminal constraints are given in the following:

\[
IV_{i,0,s} = IV_{i}^{\text{ini}} \quad \forall i, s
\]  
(4.10a)

\[
y_{m,0} = y_{m}^{\text{ini}} \quad \forall m
\]  
(4.10b)

\[
z_{mm't} = z_{mm't}^{\text{ini}} \quad \forall (m, m') \in TR, -\delta_{\text{max}} + 1 \leq t \leq -1
\]  
(4.10c)
with $\theta^\text{max} = \max \left( \max_{(m,m') \in \mathcal{TR}} \{ \theta_{mm'} \}, \max_{(m,m',m'') \in \mathcal{SQ}} \{ \bar{\theta}_{mm'm''} \} \right)$, which defines for how far back in the past the mode switching information has to be provided.

4.3.8. Total Expected Operating Cost

The total expected operating cost, $TC$, consists of the cost of purchasing electricity from contracts, the expected cost of purchasing electricity from the spot market, and the expected cost of purchasing products, as stated in the following:

$$
TC = \sum_c BC_c \sum_{t \in T} \sum_c \alpha^\text{EC}_{ct} EC_{ct} + \sum_s \phi_s \sum_{t \in T} \left( \alpha^\text{ES}_{ts} ES_{ts} + \sum_i \alpha^\text{PC}_{its} PC_{its} \right)
$$

(4.11)

where $\phi_s$ denotes the probability of scenario $s$.

4.3.9. Conditional Value-at-Risk

There are many risk measures that are widely used in practice, such as the variance of the loss distribution, shortfall probability, downside risk, and value-at-risk (VaR) (Rockafellar, 2007). The risk measure of choice in this work is the conditional value-at-risk (CVaR). For a given $\alpha \in (0, 1)$, the $\alpha$-CVaR is defined as the expected loss greater than the $\alpha$-VaR, which is the $\alpha$-quantile of the loss distribution. Since the CVaR was introduced by Rockafellar & Uryasev (2000), it has become very popular because of its ability to consider the probability density in the tail of the loss distribution, its mathematical properties from being a coherent risk measure, and the ease of incorporating it into stochastic optimization models. For more information on the use of CVaR, we refer to Rockafellar & Uryasev (2000) and Sarykalin et al. (2008). In particular, Sarykalin et al. (2008) discuss the use of VaR versus CVaR. The main insight is that the CVaR has the advantage of being sensitive to extreme tails; however, if inaccurate models are used for building the probability distributions, the VaR may be a better choice since it does not penalize outliers in the tails that may exist due to poor distributional information.

Applied to a scenario-based formulation and defined in terms of cost, the $\alpha$-CVaR corresponds to the mean cost computed over the scenarios that have costs greater than the $\alpha$-quantile. Because the model here considers price as well as demand uncertainty, we have to be cautious when defining the risk measure. If we simply apply the CVaR to the scenario set $S$, we bias toward high-demand scenar-
ios in the sense that scenarios contributing to the CVaR will most likely be associated with high product demand as illustrated in Figure 4.2a. The obvious explanation is that higher demand automatically leads to higher cost; thus, high demand is deemed to be risky and therefore contributes to the CVaR. However, higher demand also results in higher revenue, which is not considered in the CVaR defined in terms of cost; hence, high-demand scenarios are falsely regarded as unfavorable.

![Figure 4.2: CVaR defined in terms of cost](image)

![Figure 4.2: CVaR defined in terms of profit](image)

**Figure 4.2**: CVaR can be defined in terms of cost or in terms of profit, which can have different implications on the solution.

The simple solution to the problem is to define the CVaR in terms of profit by incorporating a constant revenue term, $R_s$, for each scenario. Now the $\alpha$-CVaR is defined as the expected profit computed over the scenarios that have profit values smaller than the $\alpha$-VaR, which is now the $(1 - \alpha)$-quantile of the profit distribution. As illustrated in Figure 4.2b, low-demand scenarios are likely to contribute to the CVaR because of the reduced revenues and therefore smaller profits; however, high demand may also result in low profit, e.g. if the plant does not have sufficient production capacity to satisfy the demand such that additional products have to be purchased at higher cost.

The CVaR, denoted by $CV$, is incorporated into the model by adding the following constraints to the formulation:

\[
CV = \kappa - \frac{1}{1 - \alpha} \sum_s \varphi_s \omega_s
\]

(4.12a)

\[
\kappa - \left[ R_s - \sum_c BC_c - \sum_{t \in T} \sum_c \alpha_{ct}^{EC} EC_{ct} - \sum_{t \in T} \left( \alpha_{ts}^{ES} ES_{ts} + \sum_i \alpha_{it}^{PC} PC_{its} \right) \right] \leq \omega_s \quad \forall s
\]

(4.12b)

where $\kappa$ and $\omega_s$ are continuous variables with $\kappa \in \mathbb{R}$ and $\omega_s \geq 0$. For each scenario
in which the profit is less than \( \kappa \), \( \omega_s \) takes the value of the difference between \( \kappa \) and the profit; otherwise, \( \omega_s \) is zero. When \( CV \) is maximized, it takes the value of the \( \alpha \)-CVaR. Note that in general, \( \kappa \) is not the \( \alpha \)-VaR; however, if there is one single \( \kappa \) that maximizes \( CV \), it will take the value of the \( \alpha \)-VaR when \( CV \) is maximized.

4.3.10. Objective Functions

With the proposed model, we can consider both risk-neutral and risk-averse optimization, which merely differ in their objectives. In risk-neutral optimization, where risk is not taken into account, the objective is the minimization of the total expected cost; hence, the objective function is \( TC \).

In risk-averse optimization, one tries to balance two typically conflicting objectives: optimizing the expected outcome and hedging against risk. Since the CVaR is defined as the expected profit over a small scenario subset consisting of the worst scenarios, the financial risk decreases with increasing CVaR. Therefore, the objective is to maximize a weighted sum of the total expected profit and the CVaR:

\[
\zeta TP + (1 - \zeta)CV
\]

(4.13)

with the total expected profit \( TP = \sum_s \varphi_s R_s - TC \) and \( \zeta \in [0, 1] \). Because there is usually a trade-off between expected outcome and risk in the sense that no solution can be found that maximizes both \( TP \) and \( CV \), the weighting factor \( \zeta \) can be used to specify which objective should be emphasized more. In this context, \( \zeta \) can also be seen as a parameter that sets the desired level of risk aversion. The smaller \( \zeta \), the more risk-averse is the solution, since more weight is assigned to the CVaR. Note that the revenue term, \( \sum_s \varphi_s R_s \), is constant; hence, if \( \zeta = 1 \), we obtain the risk-neutral formulation because maximizing \( TP \) is equivalent to minimizing \( TC \). If \( \zeta = 0 \), only the CVaR is maximized.

4.4. Scenario Generation and Reduction

In the proposed two-stage stochastic programming framework, each scenario corresponds to a time series, i.e. a sequence of values assigned to each time period of the scheduling horizon. Scenarios can be obtained by sampling from a suitable stochastic forecasting model. Most electricity price forecasting tools are based on univariate time series models, such as autoregressive integrated moving average (ARIMA) models (Nogales et al., 2002; Mendoza-Serrano & Chmielewski, 2014).
Aggarwal et al. (2009) present a review of methodologies for electricity price forecasting, which is not the focus of this work. This section merely provides a brief description of the ARIMA and ARIMAX (ARIMA with exogenous inputs) methods that are used to generate the scenarios in our case studies. Furthermore, we outline the scenario reduction technique that is used to reduce the scenario set to a manageable size.

### 4.4.1. Scenario Generation Using ARIMA/ARIMAX Models

An ARIMA model is a linear univariate time series model, which expresses the output at time $t$, $y_t$, as a function of observed output values in previous time periods. An ARIMA model can be formulated as follows:

\[
A(L) y_t = \frac{1}{(1 - L)^d} B(L) \epsilon_t \tag{4.14a}
\]

\[
A(L) = 1 - \phi_1 L - \phi_2 L^2 - \cdots - \phi_p L^p \tag{4.14b}
\]

\[
B(L) = 1 + \theta_1 L + \theta_2 L^2 + \cdots + \theta_q L^q \tag{4.14c}
\]

where $L$ is the lag operator, i.e. $L^k x_t = x_{t-k}$ and $(1 - L)^d x_t = x_t - x_{t-d}$.

An ARIMA model only considers time series data of the output variable. However, if we have information about exogenous inputs that are correlated with the output, we may want to make use of this information to help predicting future outputs. For instance, when predicting electricity price, useful exogenous inputs could be the temperature and the price of natural gas. The ARIMAX model is an extension to the ARIMA model that takes input time series data into account. An
ARIMAX model is given by the following formulation:

$$A(L) y_t = \frac{1}{(1 - L)^d} B(L) \epsilon_t + C(L) u_t$$  \hspace{1cm} (4.15)

with $C(L) = \eta_1 L^k + \eta_2 L^{k+1} + \eta_3 L^{k+2} + \eta_4 L^{k+b}$. Eq. (4.15) consists of the expression given in Eqs. (4.14) and an additional exogenous input term $C(L) u_t$. The nonnegative integer $k$ is the input-output lag time expressed as number of time periods. Here, the $(p + q + b)$ parameters to be estimated are $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q$, and $\eta_1, \ldots, \eta_b$.

### 4.4.2. Scenario Reduction Using Probability Distance Metrics

With an ARIMA or ARIMAX model, Monte Carlo simulation can be applied to sample scenarios. If the uncertain parameter can change significantly over time, which is certainly true in the case of spot electricity price, a large number of scenarios may be required to accurately characterize the uncertainty, resulting in a computationally intractable optimization problem. To reduce the computational effort yet still obtain good results, we seek to select a manageable number of scenarios that still represent the main features of the uncertainty. To achieve this, we apply the scenario reduction technique proposed by Dupacova et al. (2003), which selects a subset of scenarios from a given set of scenarios such that the probability distribution represented by the reduced scenario set is close to the one represented by the original scenario set. Here, the closeness of two distributions is measured in terms of a so-called probability distance.

A commonly used probability distance is the Kantorovich distance. Given two discrete probability distributions, $Q$ and $\tilde{Q}$, with scenarios $\{\omega_1, \ldots, \omega_N\}$ and $\{\tilde{\omega}_1, \ldots, \tilde{\omega}_M\}$, and probability weights $\{q_1, \ldots, q_N\}$ and $\{\tilde{q}_1, \ldots, \tilde{q}_M\}$, respectively, the Kantorovich distance between $Q$ and $\tilde{Q}$, here denoted by $D^K(Q, \tilde{Q})$, can be defined as follows:

$$D^K(Q, \tilde{Q}) = \min_{\eta_{ij}} \left\{ \sum_{i=1}^{N} \sum_{j=1}^{M} c(\omega_i, \tilde{\omega}_j) \eta_{ij} : \sum_{j=1}^{M} \eta_{ij} = q_i, \sum_{i=1}^{N} \eta_{ij} = \tilde{q}_j \forall i, j \right\}$$  \hspace{1cm} (4.16)

where $c(\omega, \tilde{\omega})$ is a nonnegative, continuous, symmetric cost function. The minimum is taken over all possible joint probabilities.

In scenario reduction, $M < N$, and the reduced scenario set $\{\tilde{\omega}_1, \ldots, \tilde{\omega}_M\}$ is a subset of the original scenario set $\{\omega_1, \ldots, \omega_N\}$. The problem then becomes which
$M$ scenarios to select and what values to assign to the new probabilities $\tilde{q}_j$ for $j = 1, \ldots, M$ such that $D^K$ is minimized. To solve the scenario reduction problem, Römisch and coworkers (Dupacova et al., 2003; Heitsch & Römisch, 2003) propose two heuristic algorithms—forward selection and backward reduction—which are fast but do not guarantee optimality. Optimality can be achieved by solving an MILP formulation recently proposed by Li & Floudas (2014). In this work, we apply the backward reduction method implemented in the scenario reduction routine SCENRED2 in GAMS (GAMS Development Corporation, 2015a).

### 4.5. Multicut Benders Decomposition

Even with a smaller number of scenarios, the optimization problem can still be very large and difficult to solve. In order to reduce the solution time, Benders decomposition (Benders, 1962) is applied. In this decomposition framework, the optimal solution is found by iteratively converging lower and upper bounds on the optimal objective function. In a minimization problem, upper bounds are obtained by fixing the first-stage variables and optimizing the second-stage decisions for each scenario. Lower bounds are obtained by solving a master problem in the space of the first-stage variables, which incorporates lower bounds on the second-stage costs. Convergence of the algorithm is achieved by improving the lower bound with the master problem which adds successively dual information of the second-stage costs obtained from the upper-bounding subproblems, which in turn potentially leads to first-stage decisions that improve the upper bound.

Lower bounds on the second-stage costs are added to the master problem in the form of cuts determined by the dual multipliers of the subproblems. In the classical Benders decomposition algorithm, one cut is generated at every iteration. To provide stronger lower bounds, Birge & Louveaux (1988) propose to generate multiple cuts at every iteration, namely one cut per scenario. This leads to a master problem that grows faster in size; however, the solution time is often shorter because fewer iterations are required. Multicut Benders decomposition has been successfully applied to various two-stage stochastic programming problems (You & Grossmann, 2013; Garcia-Herreros et al., 2014; Skar et al., 2014). For this specific problem, computational experiments have shown that multicut Benders decomposition is superior to the traditional Benders decomposition approach.

In the following, the multicut Benders decomposition approach is outlined.
Consider the original problem expressed in the following compact form:

\[
\begin{align*}
\text{min} \quad & c^T x + \sum_s \varphi_s q_s^T y_s \\
\text{s.t.} \quad & Ax \geq b \\
& T_s x + W_s y_s \geq h_s \quad \forall s \\
& y_s \geq 0 \quad \forall s
\end{align*}
\]  

(4.17)

where \(x\) denotes the vector of first-stage variables, which may be continuous or integer, and \(y_s\) are the continuous second-stage variables for scenario \(s\). To accommodate the case in which the CVaR is involved in the formulation, we simply treat \(\kappa\) as first-stage and \(\omega_s\) as second-stage variables. Parameter matrices \(A, b,\) and \(c\) are independent of the scenarios, while \(T_s, W_s, h_s,\) and \(q_s\) are scenario-specific; \(\varphi_s\) denotes the probability of scenario \(s\).

The subproblem is formulated by fixing the first-stage variables \(x\) in (4.17). After omitting the constant term in the objective function and the pure first-stage constraints \(Ax \geq b\), the dual of the subproblem at the \(k\)th iteration can be written as follows:

\[
\begin{align*}
\text{max} \quad & \sum_s (h_s - T_s x^k)^T \mu_s \\
\text{s.t.} \quad & W_s^T \mu_s \leq \varphi_s q_s \quad \forall s \\
& \mu_s \geq 0 \quad \forall s
\end{align*}
\]  

(4.18)

where \(\mu_s\) denotes the vector of dual multipliers. Note that (4.18) can be decomposed by scenario allowing solving multiple separate smaller subproblems in parallel.

The multicut master problem at the \(k\)th iteration is formulated as follows:

\[
\begin{align*}
\text{min} \quad & c^T x + \sum_s \xi_s \\
\text{s.t.} \quad & Ax \geq b \\
& (h_s - T_s x)^T \mu_s^k \leq \xi_s \quad \forall s, k
\end{align*}
\]  

(4.19)

where \(\xi_s\) is a continuous variable representing the approximate cost of scenario \(s\). So-called optimality cuts are accumulated in Eq. (4.19c) at each iteration. Note that no feasibility cuts are considered because the problem has complete recourse.

A flowchart for the multicut Benders decomposition algorithm is shown in Fig-
4. RISK-BASED PRODUCTION SCHEDULING AND ELECTRICITY PROCUREMENT

Figure 4.3. The algorithm terminates when the gap between upper and lower bounds becomes smaller than a prespecified tolerance $\epsilon$ or the time limit is reached.

4.6. Numerical Results

In the following, we apply the proposed model to an illustrative example and a real-world industrial case study. All models were implemented in GAMS 24.4.1 (GAMS Development Corporation, 2015a), and the commercial solver CPLEX 12.6.1 (IBM ILOG, 2015a) was applied to solve the MILPs on an Intel® Core® i7-2600 machine at 3.40 GHz with 8 processors and 8 GB RAM running Windows 7 Professional.

4.6.1. Illustrative Example

To demonstrate the main features of the model, we first apply it to an illustrative example in which a plant is considered that produces two products, P1 and P2. The
plant can operate in three different operating modes: off, startup, and on. The vertices of the polyhedral feasible regions of the modes are listed in Table 4.1, while the electricity consumption coefficients associated with each mode are shown in Table 4.2. The possible mode transitions are off → startup, startup → on, and on → off, for which the minimum stay times after transition are 2 h, 6 h, and 8 h, respectively. In fact, the startup process takes exactly 2 h, i.e. for the sequence off → startup → on, the fixed stay time in the startup mode is 2 h. At the start of the scheduling horizon, the plant is in the on mode. Also, it is assumed that no mode switching has occurred in the previous 8 h.

Table 4.1: Vertices associated with each operating mode of the plant from the illustrative example.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Vertex P1 [kg] P2 [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>1 0 0</td>
</tr>
<tr>
<td>startup</td>
<td>1 5 5</td>
</tr>
<tr>
<td>on</td>
<td>1 10 10</td>
</tr>
<tr>
<td></td>
<td>2 50 10</td>
</tr>
<tr>
<td></td>
<td>3 30 40</td>
</tr>
<tr>
<td></td>
<td>4 70 40</td>
</tr>
</tbody>
</table>

Table 4.2: Coefficients for linear electricity consumption correlations associated with each operating mode.

<table>
<thead>
<tr>
<th>Mode</th>
<th>δ [kWh]</th>
<th>γP1 [kWh/kg]</th>
<th>γP2 [kWh/kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>startup</td>
<td>500</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>on</td>
<td>800</td>
<td>20</td>
<td>30</td>
</tr>
</tbody>
</table>

Data regarding the inventory of each product are given in Table 4.3. We consider a two-day scheduling horizon, which starts at 8 AM of the first day. An hourly time discretization is applied, resulting in 48 time periods. One power contract is available with the following characteristics:

- Daily TOU periods: two on-peak (6 AM – 12 AM, 12 AM – 6 PM), two off-peak (0 AM – 6 AM, 6 PM – 0 AM) TOU periods
- Time-dependent price component: $20/MWh during on-peak, $15/MWh during off-peak periods
4. RISK-BASED PRODUCTION SCHEDULING AND ELECTRICITY PROCUREMENT

- Amount-dependent price component: $16/MWh if total electricity purchase less than 30 MWh, $15/MWh if between 30 MWh and 80 MWh, $14/MWh if greater than 80 MWh
- Maximum electricity purchase in every time period: 3 MWh

Table 4.3: Inventory bounds and initial inventory levels for each product.

<table>
<thead>
<tr>
<th></th>
<th>IV_{min} [kg]</th>
<th>IV_{max} [kg]</th>
<th>IV_{ini} [kg]</th>
<th>IV_{fin} [kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>600</td>
<td>6000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>P2</td>
<td>300</td>
<td>3000</td>
<td>500</td>
<td>500</td>
</tr>
</tbody>
</table>

Three sets of spot electricity price scenarios with different levels of uncertainty—low, medium, and high—are considered. Each set consists of 40 equiprobable scenarios, where each scenario is associated with a specific price profile over the 48 hours of the scheduling horizon. As an example, the price profiles along with the expected price for the price scenario set with medium level of uncertainty are shown in Figure 4.4. No limit is imposed on the amount of electricity that can be purchased from the spot market.

Figure 4.4: Spot electricity price profiles associated with the scenario set with medium level of price uncertainty.

Demand is assumed to be constant over time. This is a reasonable assumption since the products in this example are storable and small variations in demand over time can be made up by the available inventory. This assumption simplifies the generation of scenarios since no time series are needed. Also, the uncertainty can be described by considerably fewer scenarios. By further assuming that the demands for the two products are perfectly positively correlated, we generate three sets of
demand scenarios with different levels of uncertainty—low, medium, and high—with the same expected demands. The data for the three scenario sets including the probabilities for each scenario are given in Table 4.4, where $\text{Var}^D$ denotes the variance which is a measure for the level of uncertainty.

**Table 4.4**: Demand values and probabilities for each scenario of the three different demand scenario sets.

<table>
<thead>
<tr>
<th>$\bar{s}$</th>
<th>$\varphi_{\bar{s}}$</th>
<th>low $\text{Var}^D$</th>
<th>medium $\text{Var}^D$</th>
<th>high $\text{Var}^D$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$D_{P1}$ [kg/h]</td>
<td>$D_{P2}$ [kg/h]</td>
<td>$D_{P1}$ [kg/h]</td>
</tr>
<tr>
<td>1</td>
<td>0.25</td>
<td>57</td>
<td>33.25</td>
<td>54</td>
</tr>
<tr>
<td>2</td>
<td>0.50</td>
<td>60</td>
<td>35</td>
<td>60</td>
</tr>
<tr>
<td>3</td>
<td>0.25</td>
<td>63</td>
<td>36.75</td>
<td>66</td>
</tr>
</tbody>
</table>

The prices for purchasing Products P1 and P2 are $3/kg and $4/kg, respectively. The revenues for each scenario can be computed by applying the selling prices $1.5/kg and $2/kg to the demands of Products P1 and P2, respectively.

In the following, various instances are created by combining different price and demand scenario sets. Risk-neutral optimization is performed before risk-averse optimization is applied to the same instances. In the analysis of the results, we pay special attention to the assessment of the added value obtained from explicitly accounting for uncertainty in the model.

**Risk-neutral Optimization**

In risk-neutral optimization, the objective is to minimize the total expected cost. To compare the deterministic and stochastic solutions, the value of stochastic solution (VSS) is used as a measure for the difference in the solutions. Here, the VSS is defined as

$$VSS = TC_{\text{det}} - TC_{\text{sto}}$$

(4.20)

where $TC_{\text{sto}}$ is the total expected cost at the optimal solution of the two-stage stochastic problem, and $TC_{\text{det}}$ is obtained by solving the same stochastic problem with the first-stage variables fixed to the values at the optimal solution of the deterministic problem. Hence, the VSS represents the cost savings that can be expected from implementing the stochastic solution instead of the deterministic solution.
The relative VSS is defined as

$$\overline{\text{VSS}} = \frac{TC^{\text{det}} - TC^{\text{sto}}}{TC^{\text{det}}}$$

(4.21)

which is only meaningful if $TC^{\text{det}} > 0$. $TC^{\text{det}}$ could become negative if there are scenarios with negative electricity prices, which is unlikely but possible.

First, we consider three cases in which uncertainty only exists in the electricity price. The total expected costs resulting from the deterministic and stochastic solutions as well as the corresponding VSS are listed in Table 4.5. Here, $|S^D|$ and $|S^P|$ denote the numbers of demand and price scenarios, respectively. Since there is no demand uncertainty, $|S^D| = 1$. One can see that the VSS is essentially zero in the low- and medium-$\text{Var}^P$ cases. In the high-$\text{Var}^P$ case, the VSS is noticeable but very small.

**Table 4.5:** Expected costs and VSS resulting from risk-neutral optimization with only electricity price uncertainty.

| $|S^D|$ | $|S^P|$ | $\text{Var}^P$ | $TC^{\text{det}}$ [$]$ | $TC^{\text{sto}}$ [$]$ | VSS [$]$ | VSS [%] |
|-------|-------|-------------|----------------|----------------|---------|--------|
| 1     | 40    | low         | 4351           | 4351           | 0       | 0.0    |
| 1     | 40    | medium      | 4422           | 4420           | 2       | 0.0    |
| 1     | 40    | high        | 4401           | 4329           | 72      | 1.6    |

Table 4.6 shows the results for three cases in which uncertainty only exists in the product demand. Here, significant VSS can be observed, which grows with increasing level of uncertainty. For the medium-$\text{Var}^D$ case, Figure 4.5 shows the electricity purchase profiles from the deterministic and stochastic solutions. Note that the shown electricity purchase from the spot market is the expected value computed over all scenarios. The contract price comprises both the time-dependent and the amount-dependent components at the chosen purchasing amount. The comparison between the two solutions shows the impact of accounting for other scenarios besides the expected one. In the stochastic solution, the plant operates longer in order to accommodate for the high-demand scenario. By doing so, more flexibility is provided for load shifting such that the electricity consumption during the high price peak can be reduced; this in turn lowers the need for electricity purchase from the power contract.

The results from the first six cases indicate that accounting for demand uncertainty can lead to significant added value, while this does not necessarily hold true
Table 4.6: Expected costs and VSS from risk-neutral optimization with only demand uncertainty.

<table>
<thead>
<tr>
<th>$S^D$</th>
<th>$\text{Var}^D$</th>
<th>$S^P$</th>
<th>$TC^\text{det} [$]</th>
<th>$TC^\text{sto} [$]</th>
<th>VSS [$]</th>
<th>VSS [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>low</td>
<td>1</td>
<td>4528</td>
<td>4431</td>
<td>97</td>
<td>2.1</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>1</td>
<td>4674</td>
<td>4459</td>
<td>215</td>
<td>4.6</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>1</td>
<td>4967</td>
<td>4621</td>
<td>346</td>
<td>7.0</td>
</tr>
</tbody>
</table>

Figure 4.5: Electricity purchase profiles for the medium-$\text{Var}^D$ case.

for electricity price uncertainty. This raises the following question: If uncertainty exists in both electricity price and product demand, is there a benefit from considering price uncertainty in the model in addition to accounting for demand uncertainty? To answer this question, we consider nine cases involving both price and demand uncertainty. The results are shown in Table 4.7, where $|\tilde{S}^P|$ denotes the number of scenarios in the reduced scenario set, which is used in the stochastic optimization. $|\tilde{S}^P|$ indicates whether price uncertainty is considered in the model or not. Each of the nine cases is solved once only considering the expected price profile ($|\tilde{S}^P| = 1$, ignoring price uncertainty) and once with all possible price scenarios ($|\tilde{S}^P| = 40$, accounting for price uncertainty).

Table 4.7 lists some significant VSS, especially in the cases with high level of
Table 4.7: Expected costs and VSS from risk-neutral optimization with both electricity price and demand uncertainty.

<table>
<thead>
<tr>
<th>$S^D$</th>
<th>Var$^D$</th>
<th>$S^P$</th>
<th>Var$^P$</th>
<th>$\bar{S}^P$</th>
<th>$TC^{sto}$ [$]$</th>
<th>$TC^{det}$ [$]$</th>
<th>VSS [$]$</th>
<th>VSS [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>low</td>
<td>40</td>
<td>low</td>
<td>1</td>
<td>4460</td>
<td>4356</td>
<td>104</td>
<td>2.3</td>
</tr>
<tr>
<td>3</td>
<td>low</td>
<td>40</td>
<td>medium</td>
<td>1</td>
<td>4528</td>
<td>4490</td>
<td>38</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>low</td>
<td>40</td>
<td>high</td>
<td>1</td>
<td>4501</td>
<td>4455</td>
<td>46</td>
<td>0.1</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>40</td>
<td>low</td>
<td>1</td>
<td>4605</td>
<td>4413</td>
<td>192</td>
<td>4.2</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>40</td>
<td>medium</td>
<td>1</td>
<td>4670</td>
<td>4502</td>
<td>168</td>
<td>3.6</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>40</td>
<td>high</td>
<td>1</td>
<td>4637</td>
<td>4464</td>
<td>173</td>
<td>3.7</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>40</td>
<td>low</td>
<td>1</td>
<td>4898</td>
<td>4548</td>
<td>350</td>
<td>7.1</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>40</td>
<td>medium</td>
<td>1</td>
<td>4960</td>
<td>4673</td>
<td>287</td>
<td>5.8</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>40</td>
<td>high</td>
<td>1</td>
<td>4923</td>
<td>4618</td>
<td>305</td>
<td>6.2</td>
</tr>
</tbody>
</table>

In most cases, a comparison of the two instances ($|\bar{S}^P| = 1$ and $|\bar{S}^P| = 40$) shows that the impact of price uncertainty on the solution is relatively small. In other words, the added benefit from considering price uncertainty in addition to demand uncertainty is only moderate. Exceptions are the two low-Var$^D$ cases with medium and high Var$^P$.

**Risk-averse Optimization**

In risk-averse optimization, the objective is to maximize a weighted sum of the expected profit and the CVaR as given by Eq. (4.13). The VSS is defined as

$$VSS = \zeta \left( TP^{sto} - TP^{det} \right) + (1 - \zeta) \left( CV^{sto} - CV^{det} \right)$$

(4.22)

where $TP^{sto}$ and $CV^{sto}$ denote the total expected profit and CVaR at the optimal solution of the stochastic problem, and $TP^{det}$ and $CV^{det}$ are computed by solving the stochastic problem with fixed first-stage decisions obtained from the determin-
4. Risk-based Production Scheduling and Electricity Procurement

The relative VSS is defined as

$$\text{VSS} = \frac{\zeta (TP^{sto} - TP^{det}) + (1 - \zeta) (CV^{sto} - CV^{det})}{\zeta TP^{det} + (1 - \zeta) CV^{det}}$$ \hspace{1cm} (4.23)

which is only meaningful if the denominator is positive.

Risk-averse optimization is performed on the same instances presented in the previous subsection. We choose $\alpha = 0.9$ and $\zeta = 0.5$, i.e. equal weights are assigned to the total expected profit and the CVaR. The results are shown in Tables 4.8, 4.9, and 4.10. It is remarkable how much the deterministic and stochastic solutions can differ when risk is considered. The VSS reported here are considerably higher than those obtained in risk-neutral optimization. Furthermore, unlike in risk-neutral optimization, accounting for price uncertainty can lead to much improved solutions. In the cases with only electricity price uncertainty, the VSS increases dramatically with the level of uncertainty (see Table 4.8). This can also be observed in the cases in which uncertainty exists in both price and demand. Moreover, by comparing the results for $|\tilde{S}^P| = 1$ and $|\tilde{S}^P| = 40$ in each case, we see that considering electricity price uncertainty can greatly improve the solution. In fact, ignoring price uncertainty can be detrimental, to an extent that the stochastic solution is even worse than the deterministic solution (indicated by negative VSS).

<table>
<thead>
<tr>
<th>$S^D$</th>
<th>$S^P$</th>
<th>Var $P$</th>
<th>$TP^{det}$ [$]</th>
<th>$CV^{det}$ [$]</th>
<th>$TP^{sto}$ [$]</th>
<th>$CV^{sto}$ [$]</th>
<th>VSS [$]</th>
<th>VSS [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>40</td>
<td>low</td>
<td>3329</td>
<td>2981</td>
<td>3328</td>
<td>3006</td>
<td>12</td>
<td>0.4</td>
</tr>
<tr>
<td>1</td>
<td>40</td>
<td>medium</td>
<td>3258</td>
<td>2169</td>
<td>3155</td>
<td>2668</td>
<td>198</td>
<td>7.3</td>
</tr>
<tr>
<td>1</td>
<td>40</td>
<td>high</td>
<td>3278</td>
<td>1334</td>
<td>3242</td>
<td>2369</td>
<td>499</td>
<td>21.7</td>
</tr>
</tbody>
</table>

Table 4.8: Expected profits, CVaRs, and VSS from risk-averse optimization with only electricity price uncertainty.

<table>
<thead>
<tr>
<th>$S^D$</th>
<th>Var $D$</th>
<th>$S^P$</th>
<th>$TP^{det}$ [$]</th>
<th>$CV^{det}$ [$]</th>
<th>$TP^{sto}$ [$]</th>
<th>$CV^{sto}$ [$]</th>
<th>VSS [$]</th>
<th>VSS [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>low</td>
<td>1</td>
<td>3152</td>
<td>3015</td>
<td>3249</td>
<td>3058</td>
<td>70</td>
<td>2.3</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>1</td>
<td>3006</td>
<td>2631</td>
<td>3221</td>
<td>2831</td>
<td>207</td>
<td>7.4</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>1</td>
<td>2713</td>
<td>1863</td>
<td>3059</td>
<td>2384</td>
<td>434</td>
<td>19.0</td>
</tr>
</tbody>
</table>

Table 4.9: Expected profits, CVaRs, and VSS from risk-averse optimization with only demand uncertainty.
Table 4.10: Expected profits, CVaRs, and VSS from risk-averse optimization with both electricity price and demand uncertainty.

<table>
<thead>
<tr>
<th>$s^D$</th>
<th>$\text{Var}_D^P$</th>
<th>$s^P$</th>
<th>$\text{Var}_P^P$</th>
<th>$\tilde{S}^P$</th>
<th>$TP_{\text{det}}$ [$]</th>
<th>$CV_{\text{det}}$ [$]</th>
<th>$TP_{\text{sto}}$ [$]</th>
<th>$CV_{\text{sto}}$ [$]</th>
<th>VSS [$]</th>
<th>VSS [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>low</td>
<td>40</td>
<td>low</td>
<td>1</td>
<td>3220</td>
<td>2833</td>
<td>3323</td>
<td>2918</td>
<td>94</td>
<td>3.1</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>40</td>
<td>medium</td>
<td>1</td>
<td>3152</td>
<td>2072</td>
<td>3199</td>
<td>1360</td>
<td>-333</td>
<td>-12.7</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>40</td>
<td>high</td>
<td>1</td>
<td>3179</td>
<td>1252</td>
<td>3321</td>
<td>752</td>
<td>-179</td>
<td>-8.1</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>40</td>
<td>low</td>
<td>1</td>
<td>3075</td>
<td>2496</td>
<td>3296</td>
<td>2713</td>
<td>219</td>
<td>7.9</td>
</tr>
<tr>
<td>3</td>
<td>medium</td>
<td>40</td>
<td>medium</td>
<td>1</td>
<td>3010</td>
<td>1876</td>
<td>3178</td>
<td>1293</td>
<td>-207</td>
<td>-8.5</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>40</td>
<td>low</td>
<td>1</td>
<td>2782</td>
<td>1732</td>
<td>3132</td>
<td>2282</td>
<td>450</td>
<td>20.0</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>40</td>
<td>medium</td>
<td>1</td>
<td>2720</td>
<td>1332</td>
<td>3007</td>
<td>1057</td>
<td>6</td>
<td>0.3</td>
</tr>
<tr>
<td>3</td>
<td>high</td>
<td>40</td>
<td>high</td>
<td>1</td>
<td>2757</td>
<td>624</td>
<td>3164</td>
<td>635</td>
<td>209</td>
<td>12.4</td>
</tr>
</tbody>
</table>

For the medium-$\text{Var}_D^P$ medium-$\text{Var}_P^P$ case, Figure 4.6 shows the electricity purchase profiles for the stochastic solutions with $|\tilde{S}^P| = 1$ and with $|\tilde{S}^P| = 40$. In the first instance, in which price uncertainty is ignored, the solution suggests purchasing all electricity from the spot market. This solution provides maximum flexibility, and is therefore good for dealing with demand uncertainty, which is accounted for in the model. In the second instance, the model considers both price and demand uncertainty, which results in a very different solution. In this solution, more than half of the electricity is purchased from the power contract. This solution results in a slightly lower expected profit, but significantly increases the CVaR, i.e. it reduces the risk of low-profit scenarios.

Each model has up to approximately 133,000 continuous variables, 315 binary variables, and 128,000 constraints. All models were solved to zero integrality gap in less than 20 seconds wall-clock time.
4.6.2. Industrial Case Study

We now apply the proposed model to a real-world industrial case study provided by Praxair. Here, we consider a cryogenic air separation plant that produces liquid oxygen (LO2) and liquid nitrogen (LN2). In this case study, we optimize a schedule with a time horizon of one week, to which an hourly time discretization is applied resulting in 168 time periods. Two power contracts, which differ in price and availability, are considered.

To model the uncertainty in electricity price, an ARIMAX model with temperature as exogenous input is created in R (R Core Team, 2014) using data from four consecutive weeks as training data. Monte Carlo simulation is then applied to generate 1000 equiprobable price scenarios; the corresponding price profiles are shown in Figure 4.7 along with the expected price profile. Because lines are shown in the same color, the individual scenarios are indistinguishable in the diagram; however, the picture depicts the large spread in the price distribution. One can see that the level of uncertainty increases with time. We apply scenario reduction to the 1000 price scenarios and obtain a set of 50 scenarios, which are shown in Figure 4.8. Notice that the scenario reduction process assigns different probabilities to the scenarios in the reduced set.

**Figure 4.6:** Electricity purchase profiles for the medium-Var\(^D\) medium-Var\(^P\) case.
Figure 4.7: Spot electricity price profiles associated with the full set of scenarios and the expected price profile.

Figure 4.8: Spot electricity price profiles associated with the reduced scenario set.

Product demand occurs every six hours and is the total amount of product that needs to be drawn from the tank over the course of these six hours. The assumption is that there is sufficient capacity in the inventory to handle flows into and out of the inventory tank during this period of time. By further assuming that the demands for LO2 and LN2 are correlated, the uncertainty in product demand is characterized by five scenarios that resemble a normal distribution.

Again, we use the VSS to quantify the difference between the deterministic and stochastic solutions. However, note that since the stochastic optimization is performed on a reduced scenario set, but the VSS is computed over the full scenario set, only an approximate VSS (AVSS) can be obtained. The AVSS would take the value of the true VSS if the obtained first-stage decisions were the same as the ones that one would obtain from solving the stochastic problem with the full set of scenarios.
Risk-neutral vs. Risk-averse Optimization

We create three cases with different levels of demand uncertainty: low, medium, and high. For each case, two instances are created, one neglecting electricity price uncertainty ($|\hat{S}^P| = 1$), the other incorporating the 50 price scenarios from the reduced scenario set ($|\hat{S}^P| = 50$). Risk-neutral and risk-averse ($\zeta = 0.5, \alpha = 0.9$) optimization are performed on all instances for which the results are shown in Tables 4.11 and 4.12, respectively.

The most notable observation is that in risk-neutral optimization, accounting for price uncertainty does not seem to provide any added value at all. In every of the three cases, the VSS does not change when $|\hat{S}^P|$ is increased from 1 to 50. In contrast, in risk-averse optimization, there is a clear increase in VSS when price uncertainty is considered in the stochastic optimization. Note that a consistent increase in VSS can be observed although some of the stochastic problems with $|\hat{S}^P| = 50$ were not solved to optimality (see later subsection on computational results). This result is very similar to what is observed in the illustrative example, which further supports the hypothesis that accounting for electricity price uncertainty in risk-neutral optimization may be unnecessary, while in risk-averse optimization it leads to significant additional benefit.

Table 4.11: Expected costs and VSS from risk-neutral optimization.

| $|S^D|$ | $\text{Var}^D$ | $|S^P|$ | $|\hat{S}^P|$ | $TC^\text{det}$ | $TC^\text{sto}$ | AVSS | AVSS [%] |
|------|--------|------|--------|--------|--------|-----|-------|
| low  | 5      | 1    | 1000   | 12.49  | 12.49  | 0.00| 0.0   |
| medium | 5     | 1    | 1000   | 13.40  | 13.08  | 0.32| 2.4   |
| high | 5      | 1    | 1000   | 15.98  | 14.82  | 1.16| 7.2   |

In Figure 4.9, we compare for the high-$\text{Var}^D$ case the solutions obtained from the deterministic, risk-neutral, and risk-averse optimization. One can see that these three approaches lead to very different decisions. In deterministic optimization, uncertainty is ignored such that decisions are primarily driven by the differences between the power contract prices and the expected spot price. A significant amount of electricity is procured from Contract 2 because the price discount at this purchasing amount makes it less expensive than purchasing from the spot market during on-peak hours. One issue with the deterministic solution is only
Table 4.12: Expected profits, CVaRs, and VSS from risk-averse optimization.

<table>
<thead>
<tr>
<th>$S^D$</th>
<th>Var$^D$</th>
<th>$S^P$</th>
<th>$\tilde{S}^P$</th>
<th>$TP^{\text{det}}$</th>
<th>$CV^{\text{det}}$</th>
<th>$TP^{\text{sto}}$</th>
<th>$CV^{\text{sto}}$</th>
<th>AVSS</th>
<th>AVSS [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 low</td>
<td>1000</td>
<td>1</td>
<td>50</td>
<td>15.51</td>
<td>12.84</td>
<td>15.51</td>
<td>12.84</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>15.27</td>
<td>13.68</td>
<td>30.30</td>
<td>21.00</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>5 medium</td>
<td>1000</td>
<td>1</td>
<td>50</td>
<td>14.60</td>
<td>11.15</td>
<td>14.69</td>
<td>11.39</td>
<td>0.17</td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>14.80</td>
<td>11.59</td>
<td>32.17</td>
<td>23.86</td>
<td>0.32</td>
<td>2.5</td>
</tr>
<tr>
<td>5 high</td>
<td>1000</td>
<td>1</td>
<td>50</td>
<td>12.02</td>
<td>5.78</td>
<td>12.82</td>
<td>6.34</td>
<td>0.68</td>
<td>7.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>12.71</td>
<td>6.57</td>
<td>34.00</td>
<td>25.50</td>
<td>0.74</td>
<td>8.3</td>
</tr>
</tbody>
</table>

implicitly shown in the diagram, namely that there is a very high expected cost for purchasing additional products because the selected operating modes do not have sufficient production capacities in the high-demand scenarios.

Figure 4.9: Electricity purchase profiles for the high-$\text{Var}^D$ case.

In risk-neutral stochastic optimization, first-stage decisions are to a great extent driven by the need for flexibility that has to be maintained in the second stage.
4. RISK-BASED PRODUCTION SCHEDULING AND ELECTRICITY PROCUREMENT

in order to react to different scenarios. Deterministic and risk-neutral optimization lead to similar schedules for the electricity procurement from power contracts. However, the risk-neutral solution suggests selecting operating modes with higher production capacities in order to be able to accommodate high-demand scenarios. This strategy leads to considerably lower costs for purchasing additional products, especially during the last three days of the week.

The electricity procurement decisions resulting from risk-averse optimization are very different from the deterministic and risk-neutral solutions. Here, more than half of the required electricity is purchased from power contracts. Because no more electricity can be purchased from Contract 2 due to the specified purchase limit, electricity is also procured from Contract 1. Here, contracts are effectively used to hedge against the risk of low-profit scenarios. In particular, considerable amount of electricity is purchased from contracts toward the end of the week when the level of uncertainty in the spot electricity price is highest.

Impact of Product Demand Distribution

Next, the impact of different probability distributions describing the demand uncertainty is examined. We create four different demand distributions: symmetric, uniform, positively skewed, and negatively skewed. Each distribution is represented by five discrete scenarios as depicted by the figures in Table 4.13. The symmetric demand distribution is the one used in the instances presented in the previous subsection. The expected demand is the same for all four distributions.

Table 4.13: VSS in % obtained from risk-averse optimization for different demand distributions and levels of demand uncertainty.

<table>
<thead>
<tr>
<th>Demand Distribution</th>
<th>symmetric</th>
<th>uniform</th>
<th>pos. skewed</th>
<th>neg. skewed</th>
</tr>
</thead>
<tbody>
<tr>
<td>low Var(D)</td>
<td>2.1</td>
<td>2.5</td>
<td>3.5</td>
<td>1.8</td>
</tr>
<tr>
<td>medium Var(D)</td>
<td>2.5</td>
<td>3.4</td>
<td>6.0</td>
<td>1.4</td>
</tr>
<tr>
<td>high Var(D)</td>
<td>8.3</td>
<td>9.7</td>
<td>19.3</td>
<td>4.9</td>
</tr>
</tbody>
</table>

Table 4.13 shows the VSS obtained from risk-averse optimization (\(\zeta = 0.5, \alpha = 0.9\)) applied to the three cases with different levels of demand uncertainty. The highest VSS are achieved for the positively skewed demand distribution, while
the VSS are lowest for the negatively skewed distribution. Evidently, the more probability weight is assigned to low-demand scenarios, the greater is the benefit gained from stochastic optimization.

**Trade-off Between Expected Outcome and Risk**

Optimizing the expected outcome and minimizing risk are usually two conflicting objectives. In our case, it means that one cannot maximize the total expected profit and maximize the CVaR at the same time. To show this trade-off, we solve the risk-averse model with different values for the weighting factor $\zeta$. The results are shown for three different values of $\alpha$—0.8, 0.9, and 0.95—in Figure 4.10, where the CVaR is plotted against the total expected profit. The left endpoint of each curve depicts the solution of the case in which $\zeta = 0$, and the right endpoint is achieved when $\zeta = 1$. The CVaR decreases with increasing profit, which means that in order to achieve a high CVaR (reduce risk), one has to accept a lower expected profit.

![Figure 4.10: Solutions obtained by changing the weights in the objective function, showing the trade-off between total expected profit and CVaR.](image)

Note that the obtained solutions are not necessarily Pareto-optimal due to multiple reasons: (1) the stochastic problem is only solved for a subset of scenarios, (2) the problem may not be solved to optimality within the given time, and (3) since the model is nonconvex, maximizing the weighted sum does not guarantee Pareto optimality. The last limitation could be circumvented by applying the $\epsilon$-constraint approach (Hwang & Masud, 1979); however, the downside would be that the Benders decomposition algorithm could then not be used to solve the model.
**Computational Results**

With 5 demand scenarios and 50 price scenarios in the reduced scenario set, the stochastic problem is solved for 250 scenarios. Each model has approximately 3.6 million continuous variables, 3700 binary variables, and 2.7 million constraints. Here, we apply the proposed multicut Benders decomposition since the problems cannot be solved in full-space in a reasonable amount of time.

The computational results for solving the risk-neutral models and a selected set of the risk-averse models are shown in Tables 4.14 and 4.15, respectively. The wall-clock times and optimality gaps for solving each model in full-space and with multicut Benders decomposition are reported. We specify a computational budget of 7200 s (two hours). Note that the time reported for multicut Benders decomposition is the time after solving the master problem in the last iteration, which may exceed 7200 s. However, solving this last master problem is only needed for obtaining the final lower bound on the objective function and computing the final gap; the best feasible solution is obtained before solving the last master problem, at which point the computation time usually has not exceeded 7200 s. If an optimality gap is not available (n/a), it means that no feasible solution can be found within the time limit.

**Table 4.14:** Computational results for risk-neutral optimization

| $|S^D|$ | Var^D | $|S^P|$ | Full-Space | Multicut Benders |
|---|---|---|---|---|
| | | | Time [s] | Gap [%] | Time [s] | Gap [%] |
| 5 | low | 50 | 7200 | n/a | 7204 | 0.0 |
| 5 | medium | 50 | 7200 | 100 | 4871 | 0.0 |
| 5 | high | 50 | 7200 | 100 | 502 | 0.0 |

From the results in Tables 4.14 and 4.15, one can see that in almost all cases, the solver cannot even find a feasible solution within the given time if the problem is to be solved in full-space. When solving the full-space risk-neutral models for the medium- and high-Var^D cases, feasible but very poor solutions are found (100 % gap). In contrast, when the proposed multicut Benders decomposition is applied, all problems can be solved to optimality or close to optimality. In general, solving the risk-neutral models requires less computation time than solving the risk-averse models. Furthermore, the computational effort decreases with increasing level of demand uncertainty, which can be explained by the reduction of the degree of sym-
4. Risk-based Production Scheduling and Electricity Procurement

Table 4.15: Computational results for risk-averse optimization with $\zeta = 0.5$ and $\alpha = 0.9$

| $|S^D|$ | $\text{Var}^D$ | $|\bar{S}^P|$ | Demand Distribution | Full-Space Time [s] | Gap [%] | Multicut Benders Time [s] | Gap [%] |
|------|-------------|------------|------------------|---------------------|----------|---------------------------|----------|
| 5    | low         | 50         | symmetric        | 7200                | n/a      | 7531                      | 5.1      |
| 5    | medium      | 50         | symmetric        | 7200                | n/a      | 7830                      | 2.2      |
| 5    | high        | 50         | symmetric        | 7200                | n/a      | 7531                      | 0.7      |
| 5    | low         | 50         | uniform          | 7200                | n/a      | 7247                      | 3.0      |
| 5    | medium      | 50         | uniform          | 7200                | n/a      | 7499                      | 1.8      |
| 5    | high        | 50         | uniform          | 7200                | n/a      | 7550                      | 1.5      |
| 5    | low         | 50         | pos. skewed      | 7200                | n/a      | 7275                      | 5.7      |
| 5    | medium      | 50         | pos. skewed      | 7200                | n/a      | 7604                      | 2.2      |
| 5    | high        | 50         | pos. skewed      | 7200                | n/a      | 7371                      | 0.4      |
| 5    | low         | 50         | neg. skewed      | 7200                | n/a      | 7231                      | 5.2      |
| 5    | medium      | 50         | neg. skewed      | 7200                | n/a      | 7286                      | 0.9      |
| 5    | high        | 50         | neg. skewed      | 7200                | n/a      | 5165                      | 0.0      |

4.6.3. Discussion

The results from the illustrative example and the industrial case study show the difference between deterministic, risk-neutral, and risk-averse optimization, as well as the impact of electricity price and product demand uncertainty. The most remarkable insight drawn from the observations is that in risk-neutral optimization, explicitly modeling electricity price uncertainty does not lead to any significant added value. In the following, we provide an explanation for this phenomenon.

First, we notice that in the risk-neutral model, electricity prices only appear in the objective function, not in the constraints; thus, uncertainty in the price does not affect the feasible space. We can therefore restrict our analysis to the objective...
function given by Eq. (4.11), which can also be written as follows:

\[
TC = \sum_{c} BC_c + \sum_{t \in T} \sum_{c} \alpha_{ct}^{EC} EC_{ct} + \sum_{\hat{s} \in \hat{S}} \varphi_{\hat{s}}^{D} \sum_{\hat{s} \in \hat{S}^{P}} \varphi_{\hat{s}}^{P} \sum_{t \in T} \left( \alpha_{t\hat{s}}^{ES} ES_{t\hat{s}} + \sum_{i} \alpha_{it}^{PC} PC_{it\hat{s}} \right)
\]

where we distinguish between demand scenarios (\(\hat{s} \in \hat{S}^{D}\)) and price scenarios (\(\hat{s} \in \hat{S}^{P}\)), and the general scenario index \(s\) is replaced by the corresponding \((\hat{s}, \hat{s})\)-pair. The first-stage and second-stage costs are denoted by \(TC^1\) and \(TC^2\), respectively, where only the term expressing \(TC^2\) involves price scenarios.

Now we make the following assumption: Given fixed first-stage decisions and fixed product demand, the optimal solution yields the same second-stage decisions for each price scenario \(\hat{s}\), in particular the same electricity purchase from the spot market, \(ES_{t\hat{s}}\), and product purchase, \(PC_{it\hat{s}}\), i.e. \(ES_{t\hat{s}} = \bar{ES}_{ts} \forall t, \hat{s}\) and \(PC_{it\hat{s}} = \bar{PC}_{it} \forall i, t, \hat{s}\). With this assumption, \(TC^2\) can be rewritten as

\[
TC^2 = \sum_{\hat{s} \in \hat{S}^{D}} \varphi_{\hat{s}}^{D} \sum_{\hat{s} \in \hat{S}^{P}} \varphi_{\hat{s}}^{P} \sum_{t \in T} \left( \alpha_{t\hat{s}}^{ES} \bar{ES}_{ts} + \sum_{i} \alpha_{it}^{PC} \bar{PC}_{it} \right)
= \sum_{\hat{s} \in \hat{S}^{D}} \varphi_{\hat{s}}^{D} \sum_{t \in T} \left[ \sum_{\hat{s} \in \hat{S}^{P}} \varphi_{\hat{s}}^{P} \alpha_{t\hat{s}}^{ES} \bar{ES}_{ts} + \sum_{i} \alpha_{it}^{PC} \bar{PC}_{it} \right]
= \sum_{\hat{s} \in \hat{S}^{P}} \varphi_{\hat{s}}^{D} \sum_{t \in T} \left[ \mathbb{E} \left( \alpha_{t\hat{s}}^{ES} \bar{ES}_{ts} + \sum_{i} \alpha_{it}^{PC} \bar{PC}_{it} \right) \right]
\]

with \(\mathbb{E}(\alpha_{t\hat{s}}^{ES})\) denoting the expected value of \(\alpha_{t\hat{s}}^{ES}\). Since the constraints do not change for different price scenarios, Eq. (4.25) implies that under the given assumption, the stochastic model considering both demand and price uncertainty is equivalent to the model considering only demand uncertainty. In this case, both formulations will result in the same optimal first-stage decisions, which means that accounting for price uncertainty is not necessary.

The important question remains whether the assumption of equal second-stage decisions is really valid. It turns out that the assumption approximately holds true because of a particular characteristic of the electricity price uncertainty. From Figure 4.8, one can see that the electricity prices in different scenarios may differ considerably in magnitude; however, all price profiles follow essentially the same trend. In other words, the times during which the price is low or high compared to the rest of the price curve are approximately the same in all scenarios.
For fixed first-stage decisions and fixed demand (i.e. for a specific demand scenario \( \hat{s} \)), the total required electricity purchase from the spot market and additional product purchase are approximately constant, i.e.
\[
\sum_{t \in T} ES_{t\hat{s}} \approx \text{const.} \ \forall \ \hat{s}
\]
and
\[
\sum_{t \in T} PC_{it\hat{s}} \approx \text{const.} \ \forall \ i, \ \hat{s}.
\]
Then the second-stage cost only depends on the distribution of electricity and product purchases over time, which is mainly affected by the price trend rather than the price value. Hence, \( ES_{t\hat{s}} \) tend to take the same values in all price scenarios, which have almost identical price trends. The same applies trivially to \( PC_{it\hat{s}} \) since the product purchasing price does not depend on the electricity price scenario.

The analysis described above explains why no significant benefit is gained by considering electricity price uncertainty in risk-neutral optimization. However, this does not apply to risk-averse optimization since the objective function here also includes the CVaR, which only considers the most unfavorable scenarios. In this case, no equivalence between the formulation using the expected electricity price and the formulation considering different price scenarios can be deduced. In fact, the results from the case studies show that in risk-averse optimization, accounting for price uncertainty is essential for obtaining a good solution.

4.7. Summary

The work presented in this chapter addresses the simultaneous optimization of short-term production scheduling and electricity procurement under uncertainty for continuous power-intensive processes. The proposed discrete-time MILP model applies a mode-based formulation to represent the operational flexibility of the plant and a block contract formulation to model power contracts, from which electricity can be purchased besides the spot market. Two-stage stochastic programming has been applied to model both uncertainty in spot electricity price and product demand. Risk is taken into account by incorporating the CVaR into the model. Furthermore, to reduce the computational effort when solving large-scale problems, scenario reduction and multicut Benders decomposition have been applied.

An illustrative example and a real-world industrial air separation case demonstrate the capability of the proposed model and solution approach. Both risk-neutral optimization (minimization of total expected cost) and risk-averse optimization (maximization of a weighted sum of total expected profit and CVaR) have been considered. The case studies show significant differences between the solu-
tions obtained from deterministic, risk-neutral, and risk-averse optimization. Especially the electricity procurement decisions highly depend on the choice of the model. Also, in the analysis of the results, the quantification of the value of stochastic solution has been emphasized, which has led to the following remarkable insight: In risk-neutral optimization, accounting for electricity price uncertainty in the stochastic model does not result in significant additional benefit. In contrast, in risk-averse optimization, modeling price uncertainty is crucial for obtaining good solutions.
5. ROBUST SCHEDULING OF AIR SEPARATION PLANTS WITH CRYOGENIC ENERGY STORAGE

An electricity consumer’s flexibility for DSM can be significantly increased by installing capacity for storing electric energy. Here, we consider the concept of cryogenic energy storage (CES), which is to store energy in the form of liquid gas and vaporize it when needed to drive a turbine. Although CES on an industrial scale is a relatively new approach, the technology is well-known and essentially part of any air separation unit (ASU) that utilizes cryogenic separation. In this chapter, we assess the operational benefits of adding CES to an existing air separation plant. We investigate three new potential opportunities: (1) increasing the plant’s flexibility for load shifting, (2) storing purchased energy and selling it back to the market during higher-price periods, and (3) creating additional revenue by providing operating reserve capacity. We develop an MILP scheduling model and apply a robust optimization approach to model the uncertainty in reserve demand.

This chapter is organized as follows. Section 5.1 discusses the motivation for this work. In Section 5.2, the problem statement is presented. In Section 5.3, we develop a deterministic MILP scheduling model for the integrated ASU-CES plant. This model does not consider reserve market participation. Section 5.4 provides the derivation of the robust model that allows the optimization of selling reserve capacities. The industrial case study is presented in Section 5.5. In Section 5.6, we close with a summary of the main results.

5.1. Background

Energy storage is considered a key element in the efforts to enhance the efficiency and reliability of the power grid (Denholm et al., 2013; Eyer & Corey, 2010). By storing electric energy during off-peak and releasing it during on-peak hours, the need for further peak generation capacity is reduced, which averts additional capital and operating costs. Properly located storage can reduce congestion in the
transmission network; moreover, it is well-suited for providing ancillary services to
offset real-time differences between electricity supply and demand. From an elec-
tricity consumer’s point of view, energy storage can be effectively used for DSM
purposes, e.g. reducing costs by shifting load from high-price to low-price periods
(Arteconi et al., 2012).

The concept of cryogenic energy storage (CES) is to store energy in the form of
liquefied gas. When energy is needed at a later time, the liquefied gas is pumped
to high pressure and vaporized, e.g. by using low-grade heat; the high-pressure
gas can then be used to drive a turbine to generate electricity. The CES technology
is being pioneered in the UK (Chen et al., 2008; Harrabin, 2012) where the com-
pany Highview Power Storage has been running a liquid air energy storage (LAES)
pilot plant since 2011 and is currently building a higher-capacity pre-commercial
demonstration plant, which is planned to go online in the near future. Different
applications of CES have been proposed, e.g. integrating it with oxy-fuel combus-
tion and carbon capture (Pacheco et al., 2014). However, studies of CES systems at
an operational level are scarce, although essential since the true benefit of CES can
only be assessed by accounting for electricity market dynamics.

Interestingly, although CES on an industrial scale is a relatively new approach,
the technology used for CES is well-known and essentially part of every air sep-
aration unit (ASU) that utilizes cryogenic separation. In cryogenic air separation,
air is separated into its individual components at low temperatures, and typically,
large amounts of oxygen and nitrogen are liquefied to be transported to customers
via tanker trucks. Before being picked up by the trucks, the liquid products are
stored in large tanks. Hence, by simply adding a few pieces of equipment, namely
pump, heat exchanger, and turbine, one would be able to vaporize the stored liquid
products and generate electricity, which by definition makes it a CES system.

The question that we raise and try to answer in this work is whether there
are any operational benefits in adding CES capability to an existing air separation
plant. Here, we see three immediate opportunities for such an integrated ASU-CES
system: (1) In general, an ASU has a limited range of oxygen to nitrogen ratio at
which it can efficiently operate. This often leads to overproduction of one prod-
uct, which is typically vented and therefore wasted. With CES, instead of venting
overproduced products, we can store them and recover energy from them to in-
crease the plant’s flexibility for load shifting. (2) Power generated from the CES
system can be sold to the electricity market. This may be an important new source
of revenue, especially in times when the demand for LO2 and LN2 is low and the
air separation plant is therefore underutilized. (3) Similarly, the plant can also participate in the ancillary services market by providing operating reserve capacities which can be dispatched upon request. Operating reserves are required when the real-time electricity demand in the grid is higher than the supply, e.g. due to unexpected load increase or generator failures.

With its energy storage capability, an ASU-CES system does not only consume but can also generate electricity. As such, it has the opportunity to gain additional benefits by participating as a supplier in electricity markets. Just like consumers, suppliers as well are exposed to volatility in electricity price. The scheduling of power producers’ operations in the short-term electricity market is a classic problem and has been widely addressed by the power systems engineering community, in particular for hydro (Conejo et al., 2002; Lima et al., 2013) and thermal (Arroyo & Conejo, 2000; Conejo et al., 2004; Simoglou et al., 2010) power producers. Similarly, Mitra et al. (2013) optimize the scheduling of a combined heat and power plant that interacts with the electricity market as well as a chemical plant to which it supplies electricity and steam.

In general, there are two different forms in which electricity can be traded in modern electricity markets: energy and ancillary services (Kirschen & Strbac, 2004). Energy is simply what we know as electric energy or power, whereas ancillary services are backup capacities that are called upon when real-time electricity supply and demand in the grid do not match, e.g. due to equipment failures or sudden load changes. Ancillary services are therefore crucial for the stability of the power grid, and depending on how fast these capacities can be dispatched, they are categorized as operating reserve (response within minutes) or regulation (within seconds) service. Since power drawn from a CES is generated via a gas turbine, the system can react within minutes and is thus capable of providing operating reserve. Previous works have focused on the problem of deciding how much reserve is required in the grid (Wang et al., 2005; Morales et al., 2009; Xiao et al., 2011). There are very few references in the literature that address the problem from the standpoint of a reserve provider, who has to decide on the optimal amount of reserve capacity to sell. The main challenge here lies in the uncertain nature of the reserve demand. Since demand for operating reserve occurs due to unexpected contingency events, the reserve provider does not know in advance what amount and, above all, when reserve service has to be dispatched. Vujanic et al. (2012) address this issue with a robust optimization approach which has been applied to a cement plant that provides reserve by shifting load. Here, the uncer-
tainty lies in the time of required reserve dispatch and is assumed to affect the start times of the scheduled tasks.

The objective of this work is to investigate at an operational level whether adding CES capability to an existing air separation plant can be beneficial. This can be seen as an initial economic assessment of a new technology, and to the best of our knowledge, the problem as such has not been addressed in the literature before.

5.2. Problem Statement

Given an existing air separation plant that consumes air and electricity to produce gaseous oxygen (GO2), gaseous nitrogen (GN2), as well as liquid oxygen (LO2), liquid nitrogen (LN2), and liquid argon (LAr), we consider adding a CES system that can generate electricity by vaporizing LO2 or LN2. LAr will not be used for CES because it is produced in much lower quantities and has a considerably higher market value than LO2 and LN2; thus, it can be assumed to be always more profitable to sell all LAr that is produced. The power drawn from the CES can be used internally to make more products, or it can be sold to the electric energy market. Furthermore, operating reserve capacity can be provided and sold to the reserve market.

The plant has to satisfy product demand, which can be specified on an hourly basis. Gaseous product customers are connected to the plant via pipelines. We assume that there is no capacity for storing GO2 and GN2. However, there is the possibility of vaporizing LO2 and LN2 to feed the pipelines in case GO2 and GN2 production from the ASU alone is too low to satisfy the demand; in the air separation industry, this process is referred to as driox. LO2 and LN2 are stored in inventory tanks and distributed to the customers when required. The liquid product demand is considered to be the amount needed to be drawn from the tanks and shipped to the customers. Alternatively, liquid products can be purchased from third-party suppliers.

We assume that a forecast for the electricity and reserve prices is available on an hourly basis, and that the scheduling decisions do not influence these prices. The demand for operating reserve is not known in advance; however, the requested amount of power to be dispatched cannot exceed the committed reserve capacity.

The objective is to find a schedule for the integrated ASU-CES plant over a given scheduling horizon (typically a week) that minimizes the total operating cost
minus the revenue from selling power and reserve capacity. The problem is to determine for every hour of the scheduling horizon:

- the mode of operation for the ASU,
- the production level of each product,
- the amount of liquid products stored,
- the amount of liquid products used for driox,
- the amount of liquid products purchased,
- the amount of power purchased from the electricity market,
- the amount of power sold to the electricity market,
- and the reserve capacity provided.

5.3. Model Formulation

Figure 5.1 depicts the mass and power flows in an integrated ASU-CES system interacting with gas and liquid customers as well as electricity markets. Note that reserve market participation is not considered in the model described in this section but will be addressed in detail in Section 5.4. In the following, we present the proposed MILP scheduling model, which is to a large extent a special case of the model developed in Chapter 3. Unless specified otherwise, all continuous variables in this model are constrained to be nonnegative. Also, note that the nomenclature applied in this chapter differs from the one in Chapter 3.

5.3.1. Mass Balance Constraints

The ASU produces liquid and gaseous products, which are defined by the product sets \( \bar{I} \) and \( \hat{I} \), respectively, i.e. \( \bar{I} = \{LO2, LN2, LAr\} \) and \( \hat{I} = \{GO2, GN2\} \). \( \tilde{I} = \{LO2, LN2\} \) is the set of liquid products that can be stored for CES. Eq. (5.1a) states that the \( LO2 \) and \( LN2 \) produced in the ASU in each time period, denoted by \( PD_{it} \), are either stored as liquid product inventory or CES inventory. As stated in Eq. (5.1b), for liquid products not used for CES, which in this case is only \( LAr \), there is no flow into the CES tank.

\[
P_{Dit} = F_{Lit} + \bar{F}_{Lit} \quad \forall \ i \in \tilde{I}, \ t \in T
\]  (5.1a)
5. AIR SEPARATION PLANTS WITH CRYOGENIC ENERGY STORAGE

Figure 5.1: Integrated ASU-CES system. The mass and power flows are depicted by solid and dashed lines, respectively. Flow variable names are shown in parentheses.

\[ PD_{it} = FL_{it} \quad \forall \ i \in \bar{I}, \ t \in T \]  

(5.1b)

As stated in Eq. (5.2a), the liquid product inventory level at time \( t \) is the inventory level at time \( t-1 \) plus the flow into the inventory tank during time period \( t \), \( FL_{it} \), minus the amount drawn from the tank and shipped out, \( SL_{it} \), and the amount consumed by the driox process, \( VP_{it} \), during time period \( t \). Eq. (5.2b) sets lower and upper bounds on the inventory levels while Eq. (5.2c) states that the liquid demand, \( D_{it} \), has to be satisfied by the sum of the amount drawn from the inventory tank and the amount purchased, \( PC_{it} \).

\[
IV_{it} = IV_{i,t-1} + FL_{it} - SL_{it} - VP_{it} \quad \forall \ i \in \bar{I}, \ t \in T
\]

(5.2a)

\[
IV_i^{min} \leq IV_{it} \leq IV_i^{max} \quad \forall \ i \in \bar{I}, \ t \in T
\]

(5.2b)

\[
SL_{it} + PC_{it} = D_{it} \quad \forall \ i \in \bar{I}, \ t \in T
\]

(5.2c)

Change in the CES inventory level, \( \bar{IV}_t \), over time is described by the following
5. Air Separation Plants with Cryogenic Energy Storage

Constraints:

\[
\overline{FL}_{it}^{ag} = \sum_{i' \in I} \overline{ML}_{i't} \quad \forall t \in T \tag{5.3a}
\]

\[
\overline{IV}_t = \overline{IV}_{t-1} + \overline{FL}_{it}^{ag} - \frac{ED_t}{\eta} \quad \forall t \in T \tag{5.3b}
\]

\[
\overline{IV}^{\min} \leq \overline{IV}_t \leq \overline{IV}^{\max} \quad \forall t \in T \tag{5.3c}
\]

where we aggregate the flows into the CES tank, \(\overline{FL}_{it}\), assuming that LO2 and LN2 will yield the same amount of energy when vaporized and sent through a turbine. This is a fairly reasonable assumption since the two gases have similar heat capacities and heats of vaporization. As given by Eq. (5.3b), the resulting aggregate flow, \(\overline{FL}_{it}^{ag}\), minus the amount of liquid used for electricity generation, \(ED_t/\eta\), constitute the change in the CES inventory level over time period \(t\). Here, \(ED_t\) denotes the electricity discharged in time period \(t\), and \(\eta\) is the power generation efficiency given in units of power per mass. Eq. (5.3c) sets bounds on the CES inventory level.

The following equation states the mass balance for the gaseous products:

\[
PD_{it} + \rho_{i'}VP_{i't} = D_{it} + VT_{it} \quad \forall i \in \hat{I}, i' = f_{VP}^{-1}(i), t \in T \tag{5.4}
\]

where \(VT_{it}\) is the gas vented due to overproduction. With the variable \(VP_{it}\), we account for the liquid products sent to diox to be vaporized. Here, \(f_{VP} : \hat{I} \to \bar{I}\) denotes the unique mapping of gaseous to liquid products, e.g. \(f_{VP}(GO2) = LO2\).

5.3.2. Energy Balance Constraints

As stated in Eq. (5.5a), the amount of power discharged from the CES in time period \(t\), \(ED_t\), can be used internally in the ASU (\(ED_t^{in}\)) or sold to the electric energy market (\(ED_t^{en}\)). Again, note that operating reserve (\(ED_t^{re}\)) is not considered at this point. \(ED_t\) is limited by an upper bound, \(ED^{max}\), given in Eq. (5.5b), which in practice depends on the size of the turbine. In Eq. (5.5c), \(EC_t\) denotes the amount of power consumed by the ASU in time period \(t\), which consists of \(ED_t^{in}\) and \(EP_t\), the amount of power purchased from the energy market.

\[
ED_t = ED_t^{in} + ED_t^{en} \quad \forall t \in T \tag{5.5a}
\]

\[
ED_t \leq ED^{max} \quad \forall t \in T \tag{5.5b}
\]

\[
EC_t = ED_t^{in} + EP_t \quad \forall t \in T \tag{5.5c}
\]
5. AIR SEPARATION PLANTS WITH CRYOGENIC ENERGY STORAGE

5.3.3. ASU Scheduling Model

The CRS model for the ASU is represented by the following set of constraints:

\[ PD_{it} = \sum_{m \in R_m} \sum_{r \in R_m} PD_{mrit} \quad \forall \, i, \, t \in T \] (5.6a)

\[ PD_{mrit} = \sum_{j \in J_{mr}} \lambda_{mrjt} v_{mrji} \quad \forall \, m, \, r \in R_m, \, i, \, t \in T \] (5.6b)

\[ \sum_{j \in J_{mr}} \lambda_{mrjt} = \bar{y}_{mrt} \quad \forall \, m, \, r \in R_m, \, t \in T \] (5.6c)

\[ EC_t = \sum_{m \in R_m} \sum_{r \in R_m} \left( \delta_{mr} \bar{y}_{mrt} + \sum_i \gamma_{mri} PD_{mrit} \right) \quad \forall \, t \in T \] (5.6d)

\[ y_{mt} = \sum_{r \in R_m} \bar{y}_{mrt} \quad \forall \, m, \, t \in T \] (5.6e)

\[ \sum_m y_{mt} = 1 \quad \forall \, t \in T \] (5.6f)

\[ y_{mt} \in \{0, 1\} \quad \forall \, m, \, t \in T \] (5.6g)

\[ \bar{y}_{mrt} \in \{0, 1\} \quad \forall \, m, \, r \in R_m, \, t \in T \] (5.6h)

where \( R_m \) is the set of convex regions associated with mode \( m \), and \( J_{mr} \) is the set of vertices of region \( r \in R_m \). The binary variable \( y_{mt} \) equals 1 if mode \( m \) is selected in time period \( t \), and \( \bar{y}_{mrt} \) equals 1 if region \( r \in R_m \) is selected in time period \( t \).

The amount of product \( i \) produced in time period \( t \) is denoted by \( PD_{it} \). Associated with \( PD_{it} \) is the disaggregated variable \( PD_{mrit} \) for mode \( m \) and region \( r \), which is expressed as a convex combination of the corresponding vertices, \( v_{mrji} \). The amount of electricity consumed, \( EC_t \), is a linear function of \( PD_{it} \) with a constant \( \delta_{mr} \) and coefficients \( \gamma_{mri} \) specific to the selected mode and region.

The transition constraints are the same as the ones presented in Chapter 4, but are shown here again for the sake of completeness:

\[ \sum_{m' \in TR_m} z_{m'm,t-1} - \sum_{m' \in \overline{TR}_m} z_{mm',t-1} = y_{mt} - y_{m,t-1} \quad \forall \, m, \, t \in T \] (5.7a)

\[ y_{mt'} \geq \sum_{k=1}^{\theta_{mm'}} z_{mm',t-k} \quad \forall \, (m, m') \in TR, \, t \in T \] (5.7b)

\[ z_{mm',t-\theta_{mm'm''}} = z_{mm',t} \quad \forall \, (m, m', m'') \in SQ, \, t \in T \] (5.7c)

\[ z_{mm',t} \in \{0, 1\} \quad \forall \, (m, m') \in TR, \, t \in T \] (5.7d)

where \( TR_m = \{m' : (m', m) \in TR\} \) and \( \overline{TR}_m = \{m' : (m, m') \in TR\} \) with \( TR \) be-
ing the set of all possible mode-to-mode transitions, $SQ$ is the set of predefined sequences, $\theta_{mm'}$ is the minimum stay time in mode $m'$ after switching to it from mode $m$, and $\bar{\theta}_{mm'm''}$ is the fixed stay time in mode $m'$ in the corresponding sequence. The binary variable $z_{mm't}$ equals 1 if and only if the plant switches from mode $m$ to mode $m'$ at time $t$.

5.3.4. Boundary Conditions

The following initial conditions set the initial inventory levels, the initial operating mode, and the mode switching history:

\begin{align}
IV_{i,0} &= IV^{ini}_i \quad \forall \ i \in \tilde{I} \\
\overline{IV}_0 &= \overline{IV}^{ini} \\
y_{m,0} &= y^{ini}_m \quad \forall \ m \\
z_{mm't} &= z^{ini}_{mm't} \quad \forall \ (m,m') \in TR, \ -\theta^{max} + 1 \leq t \leq -1 \quad (5.8c)
\end{align}

with $\theta^{max} = \max \left( \max_{(m,m') \in TR} \{\theta_{mm'}\}, \max_{(m,m',m'') \in SQ} \{\bar{\theta}_{mm'm''}\} \right)$.

In the following terminal constraints, we simply set lower bounds on the final inventory levels:

\begin{align}
IV_{i,t} &\geq IV^{fin}_i \quad \forall \ i \in \tilde{I} \quad (5.9a) \\
\overline{IV}_t &\geq \overline{IV}^{fin} \\
\end{align}

5.3.5. Objective Function

The objective is to minimize the total net operating cost $TC$. As stated in Eq. (5.10), $TC$ is defined as the sum of the electricity cost, the product purchase cost, and the diox cost minus the revenue from selling power to the energy market over the entire scheduling horizon:

\[
TC = \sum_{t \in T} \left[ \alpha^EP_t P_t + \sum_{i \in I} \alpha^{PC}_{it} PC_{it} + \sum_{i \in I} \alpha^{VP}_{it} VP_{it} - \mu \alpha^EP_t ED^n_t \right] 
\]

where $\mu$ is a parameter between 0 and 1. It is assumed that selling power to the grid comes with a small additional cost, e.g., transaction cost, such that the sales price is slightly lower than the price at which we buy power from the market.
5.4. Robust Model with Reserve Market Participation

As mentioned, the model presented in the previous section does not consider the potential interaction with the operating reserve market. In this section, we develop a model that allows the participation of an ASU-CES plant in the reserve market by applying a robust optimization approach to account for the uncertainty in reserve demand.

In general, one distinguishes between spinning and non-spinning reserves. Generation resources providing spinning reserve have to be already online, i.e. synchronized with the system, when reserve is requested. Non-spinning reserve can be provided by generators not synchronized with the grid, but capable of starting up and serving demand within a given time frame. In the electricity market operated by PJM Interconnection LLC (PJM), which is the one considered in the case study in Section 4.6.2, the required response time is ten minutes (PJM Interconnection LLC, 2014b). An ASU-CES plant is capable of providing both spinning and non-spinning reserve. However, since the reward for providing spinning reserve is typically significantly higher than for providing non-spinning reserve (PJM Interconnection LLC, 2014c), we consider the former in this model.

Selling reserve capacity is attractive because the reserve provider is rewarded even when no actual generation of power is required. Whenever reserve service is actually dispatched, a payment is made to the provider in addition to the reward for its committed reserve capacity. This market incentive reflects the value of flexible generation resources that can react quickly to unexpected changes in the power grid. However, there is an inherent risk associated with providing reserve service because one does not know in advance when reserve will be required. Since non-compliance would result in prohibitively high penalties, reserve providers have to operate in a way such that dispatch of the committed reserve capacities can be guaranteed.

In principle, we can incorporate participation in the reserve market by extending the model presented in Section 5.3 to the following formulation:

\[
\min \quad TC - \sum_{t \in T} \alpha_t re ED_t re - \sum_{t \in T} \alpha_t RC_t \quad (5.11a)
\]

\[
\text{s.t.} \quad \text{Eqs. (5.1), (5.2), (5.3a), (5.4), (5.5c), (5.6)–(5.8), (5.9a)}
\]

\[
ED_t = ED_t iu + ED_t en + ED_t re \quad \forall \ t \in T \quad (5.11b)
\]

\[
ED_t \leq ED_{max} \quad \forall \ t \in T \quad (5.11c)
\]
5. AIR SEPARATION PLANTS WITH CRYOGENIC ENERGY STORAGE

\[ \overline{IV}_t = \overline{IV}_{t-1} + \frac{P^\text{ng}_{t}}{\eta} - \frac{ED_t}{\eta} \quad \forall \ t \in \mathcal{T} \]  (5.11d)

\[ \overline{IV}^\text{min} \leq \overline{IV}_t \leq \overline{IV}^\text{max} \quad \forall \ t \in \mathcal{T} \]  (5.11e)

\[ RC^\text{min} x_t \leq RC_t \leq ED^\text{max} x_t \quad \forall \ t \in \mathcal{T} \]  (5.11f)

\[ ED_t \geq ED^\text{min} x_t \quad \forall \ t \in \mathcal{T} \]  (5.11g)

\[ ED_{t-1} \geq ED^\text{min} x_t \quad \forall \ t \in \mathcal{T} \]  (5.11h)

\[ x_t \in \{0, 1\} \quad \forall \ t \in \mathcal{T} \]  (5.11i)

where \( ED_t^\text{re} \) is the amount of power dispatched as reserve and \( RC_t \) is the reserve capacity provided in time period \( t \), \( \alpha_t^\text{re} \) and \( \alpha_t^\text{RC} \) are the corresponding unit prices.

As shown in Eq. (5.11b), part of the power discharged from the CES is \( ED_t^\text{re} \). The binary variable \( x_t \) in Eq. (5.11f) equals 1 if reserve capacity is provided in time period \( t \). \( RC^\text{min} \) is the minimum amount of reserve capacity to be provided in order to take part in the reserve market. The upper bound for \( RC_t \) is given by the physical capacity of the discharging system, \( ED^\text{max} \). In order to be able to respond to a spinning reserve request at any time within a time period, the generator has to be online during that time period as well as the previous time period. These constraints are stated in Eqs. (5.11g) and (5.11h).

Here, \( ED_t^\text{re} \), the reserve demand, is the uncertain parameter, which not only is unknown a priori, but is bounded above by \( RC_t \), i.e. the uncertainty depends on how much reserve capacity we decide to provide. In robust optimization, the uncertainty is specified in terms of an uncertainty set from which any point is a possible realization of the uncertainty. The goal is to find a solution that is feasible for all possible realizations of the uncertainty while minimizing (or maximizing) the objective function. For further reading on robust optimization theory and its applications, we refer to comprehensive reviews in the literature (Ben-Tal et al., 2009; Bertsimas et al., 2011; Gabrel et al., 2014). In the following, we construct the appropriate uncertainty set for our problem and derive the corresponding robust model.

5.4.1. Uncertainty Set

Given a committed reserve capacity in time period \( t \), \( RC_t \), \( ED_t^\text{re} \) can take any values between zero and \( RC_t \). Hence, the uncertainty set can simply be formulated as

\[ U(RC) = \{ ED_t^\text{re} : 0 \leq ED_t^\text{re} \leq RC_t \quad \forall \ t \in \mathcal{T} \} \]  (5.12)
where $RC = [RC_1, RC_2, \ldots, RC_T]^T$ and $ED_{\text{re}} = [ED_{\text{re}}^1, ED_{\text{re}}^2, \ldots, ED_{\text{re}}^T]^T$; $U(RC)$ indicates that the uncertainty set is a function of $RC$.

Optimization considering box uncertainty sets such as the one given in Eq. (5.12) has been considered in the early works of Soyster (1973) and of Friedman & Reklaitis (1975b). However, considering such an uncertainty set is often too conservative. In our particular case, we would solve the model for the worst case in which the maximum amount of reserve is requested in every time period, i.e. $ED_{\text{re}}^t = RC_t \forall t \in T$. According to EnerNOC (2014), reserve dispatch is requested from an individual reserve provider 5 to 25 times a year. The worst case described above is therefore highly unlikely.

To reduce the level of conservatism, we adopt the “budget of uncertainty” approach introduced by Bertsimas & Sim (2004). By defining the normalized reserve demand $w_t = ED_{\text{re}}^t / RC_t$, we can restrict the size of the uncertainty set by defining it as follows:

$$U(RC) = \left\{ w : ED_{\text{re}}^t = RC_t w_t, \ 0 \leq w_t \leq 1 \ \forall t \in T, \ \sum_{t \in T} w_t \leq \Gamma \right\}$$

where $w = [w_1, w_2, \ldots, w_T]^T$. The idea is to set a limit on the total reserve demand in any realization of the uncertainty. The size of the uncertainty set and therefore the level of conservatism increase with $\Gamma$, and for $\Gamma = |T|$, the uncertainty sets given in Eqs. (5.13) and (5.12) are equivalent.

By changing $\Gamma$, we can adjust the level of conservatism. Yet this uncertainty set formulation is still too conservative because it puts too much emphasis on the first time periods. We illustrate this point with the following example: Say we have 9 time periods and we decide to provide the same amount of reserve capacity in every time period. With $\Gamma$ set to 3, we effectively restrict the number of time periods in which the maximum amount of reserve may be dispatched to 3. As illustrated in Figure 5.2, one of the possible realizations is the maximum reserve dispatch in time periods 1 to 3, which is usually the worst-case scenario. However, this case again is not realistic and would lead to an unnecessarily conservative solution.

As opposed to occurring consecutively at the beginning of the time horizon, it is more realistic that reserve events are spread out over the entire time horizon. We propose to incorporate this insight by applying the following uncertainty set:

$$U(RC) = \left\{ w : \left( ED_{\text{re}}^t = RC_t w_t, \ 0 \leq w_t \leq 1 \ \forall k \in K_t, \ \sum_{k=1}^{t} w_k \leq \Gamma_t \right) \ \forall t \in T \right\}$$
Figure 5.2: For the illustrative example, the diagram shows a possible realization of the uncertainty if the uncertainty set given by Eq. (5.13) with $\Gamma = 3$ is applied. 

where a budget parameter, $\Gamma_t$, is defined for each $t$ and applied to time periods $k \in K_t$ with $K_t = \{1,2,\ldots,t\}$. For this uncertainty set to have the desired effect, $\Gamma_t$ has to be monotonically increasing with $t$. We illustrate this by applying it to the same example as before: Again, we have 9 time periods and provide the same amount of reserve capacity in all time periods. Now we set $\Gamma = [1,1,1,2,2,2,3,3,3]^T$, which means that maximum reserve request can occur once in any of the first 3 time periods, twice in the first 6 time periods, and three times in the entire 9-period horizon. A possible realization is shown in Figure 5.3, which is more realistic compared to the case depicted in Figure 5.2.

Figure 5.3: For the illustrative example, the diagram shows a possible realization of the uncertainty if the uncertainty set given by Eq. (5.14) with $\Gamma = [1,1,1,2,2,2,3,3,3]^T$ is applied.

5.4.2. Robust Counterpart

Based on the chosen uncertainty set defined in Eq. (5.14), we derive the robust model, which in the robust optimization literature is referred to as the robust coun-
terpart. First, we express the CES inventory level in closed form:

\[
\overline{IV}_t = \overline{IV}_0 + \sum_{k=1}^{t} \left[ FL_{k}^{ag} - \frac{1}{\eta} \left( ED_{k}^{in} + ED_{k}^{en} + ED_{k}^{re} \right) \right]
\]

\[
= \overline{IV}_0 + \sum_{k=1}^{t} \left[ FL_{k}^{ag} - \frac{1}{\eta} \left( ED_{k}^{in} + ED_{k}^{en} \right) \right] - \frac{1}{\eta} \sum_{k=1}^{t} ED_{k}^{re}
\]

(5.15)

In this way, the uncertain parameter \( ED_{k}^{re} \) appears in a summation over \( k \in K_t \), as in the formulation of the uncertainty set.

By considering the proposed uncertainty set and applying the worst-case approach to each constraint, we arrive at the following formulation:

\[
\begin{align*}
\min & \quad TC - \min_{w \in U(RC)} \left\{ \sum_{t \in T} \alpha_{t}^{w} RC_t w_t \right\} - \sum_{t \in T} \alpha_{t}^{RC} RC_t \\
\text{s.t.} & \quad \text{Eqs. (5.1), (5.2), (5.3a), (5.4), (5.5c), (5.6)–(5.8), (5.9a)} \\
& \quad ED_{t}^{in} + ED_{t}^{en} + \max_{w \in U(RC)} \left\{ RC_t w_t \right\} \leq ED_{\max} \quad \forall t \in \overline{T} \\
& \quad \overline{IV}_0 + \sum_{k=1}^{t} u_k - \frac{1}{\eta} \min_{w \in U(RC)} \left\{ \sum_{k=1}^{t} RC_k w_k \right\} \leq \overline{IV}_{\max} \quad \forall t \in T \\
& \quad - \overline{IV}_0 - \sum_{k=1}^{t} u_k + \frac{1}{\eta} \max_{w \in U(RC)} \left\{ \sum_{k=1}^{t} RC_k w_k \right\} \leq -\overline{IV}_{\min} \quad \forall t \in \overline{T} \\
& \quad RC_{\min} x_t \leq RC_t \leq RC_{\max} x_t \quad \forall t \in \overline{T} \\
& \quad ED_{t}^{in} + ED_{t}^{en} + \min_{w \in U(RC)} \left\{ RC_t w_t \right\} \geq ED_{\min} x_t \quad \forall t \in T \\
& \quad ED_{t-1}^{in} + ED_{t-1}^{en} + \min_{w \in U(RC)} \left\{ RC_{t-1} w_{t-1} \right\} \geq ED_{\min} x_t \quad \forall t \in \overline{T} \\
\end{align*}
\]

(5.16a)–(5.16h)

which can be interpreted as a bilevel problem. The lower-level problems in Eqs. (5.16a), (5.16b), (5.16c), (5.16f), and (5.16g) are trivially solved. Given \( RC \), the solutions to \( \min_{w \in U(RC)} \left\{ \sum_{t \in T} \alpha_{t}^{w} RC_t w_t \right\} \), \( \min_{w \in U(RC)} \left\{ \sum_{k=1}^{t} RC_k w_k \right\} \), and \( \min_{w \in U(RC)} \left\{ RC_t w_t \right\} \) are simply zero, while the solution to \( \max_{w \in U(RC)} \left\{ RC_t w_t \right\} \) is \( RC_t \). The lower-level problem in Eq. (5.16d), however, requires special treatment.

To reformulate Eq. (5.16d), we formulate the following auxiliary problem for each \( t \in \overline{T} \):

\[
\max \sum_{k=1}^{t} RC_k w_k 
\]

(5.17a)
The dual of the auxiliary problem is:

\[
\begin{align*}
\text{min} & \quad \Gamma_t q_t + \sum_{k=1}^t s_{tk} \\
\text{s.t.} & \quad q_t + s_{tk} \geq RC_k \quad \forall k \in K_t \\
& \quad q_t \geq 0 \\
& \quad s_{tk} \geq 0 \quad \forall k \in K_t
\end{align*}
\]

By strong duality, since Problem (5.17) is feasible and bounded for all \(\Gamma_t \in [0,t]\), the dual problem (5.18) is also feasible and bounded, and moreover, (5.17) and (5.18) have the same optimal objective function value. Since Problem (5.18) is a minimization problem and every feasible solution will yield an objective function value equal to or greater than the minimum, we can substitute (5.18) for (5.17) in the robust counterpart formulation. The optimization will automatically drive the objective function value of (5.18) to its minimum, which coincides with the solution of Problem (5.17).

By replacing the lower-level problems with their respective solutions, the robust counterpart in (5.16) can be reformulated into the following MILP:

\[
\begin{align*}
\text{min} & \quad TC - \sum_{t \in \mathcal{T}} \alpha_t^{RC} RC_t \\
\text{s.t.} & \quad \text{Eqs. (5.1), (5.2), (5.3), (5.4), (5.5), (5.6), (5.7), (5.8), (5.9)} \\
& \quad ED_t^{\text{in}} + ED_t^{\text{en}} + RC_t \leq ED_t^{\text{max}} \quad \forall t \in \mathcal{T} \\
& \quad \overline{V}_0 + \sum_{k=1}^t u_k \leq \overline{V}_0^{\text{max}} \quad \forall t \in \mathcal{T} \\
& \quad -\overline{V}_0 - \sum_{k=1}^t u_k + \frac{1}{\eta} \left( \Gamma_t q_t + \sum_{k=1}^t s_{tk} \right) \leq -\overline{V}_0^{\text{min}} \quad \forall t \in \mathcal{T} \\
& \quad q_t + s_{tk} \geq RC_k \quad \forall t \in \mathcal{T}, k \in K_t \\
& \quad q_t \geq 0, s_{tk} \geq 0 \quad \forall t \in \mathcal{T}, k \in K_t \\
& \quad RC_t^{\text{min}} x_t \leq RC_t \leq ED_t^{\text{max}} x_t \quad \forall t \in \mathcal{T} \\
& \quad ED_t^{\text{in}} + ED_t^{\text{en}} \geq ED_t^{\text{min}} x_t \quad \forall t \in \mathcal{T} \\
& \quad ED_{t-1}^{\text{in}} + ED_{t-1}^{\text{en}} \geq ED_{t-1}^{\text{min}} x_t \quad \forall t \in \mathcal{T}
\end{align*}
\]
Note that the variable $RC_t$, which affects the uncertainty set, appears linearly in the robust counterpart formulation.

5.5. Industrial Case Study

We now apply the proposed model to a real-world industrial case study for which the data are provided by Praxair. Given is an existing air separation plant that only produces liquid products and has to satisfy LO2 and LN2 demand. The scheduling horizon is one week to which we apply an hourly time discretization resulting in 168 time periods.

For our analysis, we choose a benchmark case in which the plant utilization is about 60%. Here, plant utilization is defined as the ratio between the liquid product demand and the maximum amount of liquid products that the plant is able to produce over the given scheduling horizon. The PJM electricity market is considered, and the hourly day-ahead energy and spinning reserve prices are taken from the week of June 23 to 29, 2014 (PJM Interconnection LLC, 2014c,a).

To set a baseline, we first determine the optimal production schedule for the existing ASU, without considering a CES add-on. The solution is shown in Figure 5.4 in the form of the power consumption profile over the entire week. By examining it in conjunction with the electricity price profile, one can clearly see the trend of operating the plant at a higher production level when electricity price is low and at a lower level when electricity price is high. The solution even suggests shutting down the plant for 54 hours in the middle of the week when the peak electricity price is the highest. During this period of time, product demand is satisfied by drawing from the inventory.

In the following, we investigate the impact of an added CES system to the optimal production schedule. In Section 5.5.1, we first consider the case of an integrated ASU-CES plant that only participates in the energy market. Here, all information is assumed to be deterministic. In Section 5.5.2, we consider the case in which the ASU-CES plant also participates in the reserve market, which requires accounting for the uncertainty in reserve demand. We show the additional benefit of providing reserve capacity and examine the impact of the specified level of conservatism. Finally, we perform a sensitivity analysis in Section 5.5.3, in which we observe the change in economic performance with different CES efficiencies and different lev-
Figure 5.4: For the benchmark case only considering the ASU, the optimal solution suggests load shifting toward low-price time periods and shutting down the plant for 54 hours in the middle of the week.

5.5.1. ASU-CES with Energy Market Participation

To the existing ASU, we add a CES system with the following specifications:

- 70% overall efficiency
- 10 MW maximum power output
- 750,000 kg CES inventory capacity

It should be noted that these numbers are chosen rather conservatively. 70% efficiency is below the 80% that, according to Chen et al. (2008), could be reached by integrating waste heat. This efficiency factor should not be mistaken for the parameter \( \eta \) in the model formulation. Rather, \( \eta = \eta^{CES} \eta^{ASU} \), where \( \eta^{CES} \) is the overall efficiency factor referred to in the CES specifications, and \( \eta^{ASU} \) is given in kWh/kg and reflects the efficiency of the ASU. 10 MW for maximum power output is also reasonable considering that the company Highview Power Storage claims to be able to design CES plants with more than 50 MW output. Finally, 750,000 kg only constitute about 10–20% of the inventory capacity of a typical industrial-scale cryogenic air separation plant. In fact, it turns out that in this case, the CES inventory capacity does not have a big impact on the solution. Except in the extreme case in which there is no liquid product demand (c.f. sensitivity analysis in Section 5.5.3), the maximum CES capacity is never reached.

We apply the model presented in Section 5.3, which considers generating electricity for internal use or for being sold to the energy market, and obtain an optimal
solution that yields a total operating cost that is 2.1% less than in the case of only operating the ASU. Figure 5.5 shows the power consumption profile, which indicates that the ASU-CES plant operates 20 hours longer compared to the ASU-only case. Evidently, the plant is utilizing more of its production capacity to benefit from the added CES capability. Figure 5.5 also shows that most of the energy consumed by the ASU is purchased from the market; however, in some time periods, in which the electricity prices are very high, the plant runs completely on power recovered from the CES.

Figure 5.5: In the benchmark case with integrated ASU and CES, the plant operates longer and part of the electricity consumed is recovered from the CES.

Figure 5.6 shows the LO2 and LN2 inventory profiles as well as the flows into the product inventory tanks and the demands, which match the flows out of the tanks. Here, the general trend is that the plant produces liquid products to build up inventory to prepare for the plant shutdown. When the ASU is shut down, products are drawn from the inventory tanks to satisfy demand until the safety stock level is reached and the plant starts producing again.

Figure 5.7 shows the change in CES inventory level as well as the flows into and out of the CES tank. One can observe that the stored energy is released during high-price hours to either power the ASU or be sold to the energy market.

5.5.2. ASU-CES with Energy and Reserve Market Participation

We now apply the robust model developed in Section 5.4 to consider reserve market in addition to energy market participation. Moreover, we explore the flexibility of adjusting the level of conservatism by setting the budget parameters $\Gamma_t$. We consider the following three cases:

- Case 1: $\Gamma_t = 1 \forall t \in \mathcal{T}$, i.e. reserve is requested at most once during the week.
5. Air Separation Plants with Cryogenic Energy Storage

Figure 5.6: The change in product inventory levels shown in the diagrams is due to the flows into and out of the inventory tanks.

Figure 5.7: Liquid flow into the CES tank increases the CES inventory while it depletes when stored liquid is converted to power either to be used internally or sold to the market.

- Case 2: $\Gamma_t = 1$ for $t = 1, \ldots, 48$, $\Gamma_t = 2$ for $t = 49, \ldots, 96$, $\Gamma_t = 3$ for $t = 97, \ldots, 168$, i.e. reserve can be requested once during the first 2 days, twice during the first 4 days, and three times during the entire week.

- Case 3: $\Gamma_t$ increases every 24 time periods by 1, i.e. reserve could be requested once every day.

In Case 1, a realistic assumption on the uncertainty is made but may be considered slightly risky. Case 2 is already very conservative by assuming that reserve could
be requested up to three times during one week. The uncertainty assumption in Case 3 is even more conservative.

Solving the model for Case 1 yields the power consumption profile shown in Figure 5.8. Comparing this diagram to the one in Figure 5.5, we can see that the total power consumption is similar; however, more power is recovered from CES for internal use. This stems from the requirement of running the CES discharging system at a minimum level whenever spinning reserve is provided. Thus, the amount of power recovered from CES now does not only depend on the energy but also the reserve price. This effect can be observed in Figure 5.9, which shows that whenever reserve capacity is provided, power is discharged from the CES.

**Figure 5.8:** In Case 1, by participating in the reserve market, more power is recovered from CES for internal use.

Figure 5.9 shows the CES inventory profile resulting from the scenario in which no reserve power is dispatched. In each time period in which reserve capacity is provided, the CES inventory level ensures that the maximum amount of reserve can be dispatched. As a result, the final CES inventory level depicted here is not zero but has the value of the highest reserve capacity provided during any hour of the week. Figure 5.9 also shows the spinning reserve price so that we can see that reserve is provided during high-price hours. The optimal solution of Case 1 results in a total operating cost that corresponds to another 9.1% reduction compared to the case in which only participation in the energy market is possible. Compared to the ASU-only case, this constitutes a quite significant cost reduction of 11.0%.

Figure 5.10 shows the CES inventory profiles and the flows into and out of the CES tank for Case 2 and Case 3. It is remarkable that the solutions suggest to provide almost the same amount of reserve capacity as in Case 1 although the uncertainty assumptions are much more conservative. However, more inventory has to be kept in the CES tank in order to guarantee feasible reserve dispatch, which
5. AIR SEPARATION PLANTS WITH CRYOGENIC ENERGY STORAGE

Figure 5.9: For Case 1, the CES inventory profile is shown for the scenario in which no reserve power is dispatched. Reserve capacity is provided during time periods in which the spinning reserve price is high.

is indicated by the final CES inventory levels. In Case 2, the final inventory level is three times as high as in Case 1, while in Case 3, it is seven times as high as in Case 1. As a result, the cost savings are less; however, we still achieve 7.0% and 2.8% cost reduction in Case 2 and Case 3, respectively, compared to the case with no reserve market participation.

5.5.3. Sensitivity Analysis

In the benchmark case, assuming the Case 2 level of conservatism, a total operating cost reduction of 8.9% has been achieved by adding CES capability to the existing ASU and participating in the energy market as well as the reserve market. This number can vary considerably under different conditions. In this respect, the two main parameters are CES efficiency and plant utilization. We perform a sensitivity analysis by varying these parameters and determining the cost benefit of an integrated ASU-CES plant compared to an ASU-only plant for each case. The results are presented in Figure 5.11, which shows for different overall efficiency and utilization factors the relative cost savings over the given time horizon of a week.

One can see that the cost savings increase with increasing overall efficiency and decreasing plant utilization. Here, plant utilization can be seen as a measure for available flexibility. The lower the utilization is, the more flexible the plant is in its operations, which allows it to take more advantage of the added CES capability. For instance, for an overall efficiency of 80%, we achieve a cost reduction of 12.1% and 22.6% per week for a plant utilization of 60% and 40%, respectively.
Figure 5.10: CES inventory levels for Case 2 (top diagram) and Case 3 (bottom diagram).

Figure 5.11: In general, relative cost savings increase with increasing efficiency and decreasing utilization.

Essentially, these results imply that investing in a CES system may be especially worthwhile for an air separation plant that is often underutilized. Note that the case of zero utilization corresponds to operating a stand-alone CES plant for the sole purpose of selling energy and operating reserve.
5. Air Separation Plants with Cryogenic Energy Storage

5.5.4. Computational Results

All models are implemented in GAMS 24.2.3 (GAMS Development Corporation, 2014), and the commercial solver CPLEX 12.5 (IBM ILOG, 2014) has been applied to solve the MILPs. The largest model solved has 69,311 continuous variables, 4296 binary variables, and 30,597 constraints. All models have been solved to zero integrality gap in less than 60 seconds on an Intel® Core™ i7-2600 machine at 3.40 GHz with 8 processors and 8 GB RAM running Windows 7 Professional.

5.6. Summary

The work presented in this chapter has assessed at an operational level the economic benefits of adding a CES system to an existing cryogenic air separation plant. We have developed an MILP scheduling model for an integrated ASU-CES plant that incorporates the possibility of recovering energy from CES for internal use or for being sold to the electric energy market. Using a robust optimization approach, this model has been further extended to consider uncertainty in operating reserve demand. This allows the model to consider reserve market participation and yield solutions that guarantee reserve dispatch feasibility under the committed reserve capacity. Furthermore, budget parameters are used to adjust the level of conservatism in the solution.

The proposed model has been applied to a real-world industrial case study. The results exhibit typical relative cost savings of approximately 10% under relatively conservative efficiency and uncertainty assumptions. A sensitivity analysis shows that besides the CES efficiency, economic benefits strongly depend on the level of plant utilization. If the level of utilization is low, which allows high flexibility for load shifting, cost reduction up to over 20% (for 40% utilization) can be achieved. This suggests that an added CES system may be an especially good option for underutilized air separation plants.

This work has provided an initial economic assessment of an integrated ASU-CES system. Further investigations are necessary, especially on capital costs, in order to determine the economic feasibility of this approach. We have considered the CES as an add-on to an already existing ASU. However, a more efficient integration of ASU and CES may require a complete redesign of the air separation plant. For example, the efficiency may be increased by storing liquid air as energy, as opposed to LO2 and LN2.
6. Adjustable Robust Scheduling of Continuous Industrial Processes Providing Interruptible Load

To ensure the stability of the power grid, operating reserves are called upon when electricity supply does not meet demand due to unexpected changes in the grid. As discussed in Chapter 5, operating reserve can be provided by electricity generators with relatively short ramp-up times. Alternatively, operating reserve can be provided by consumers that can quickly reduce their electricity consumption; such load reduction capacity is referred to as interruptible load. In this chapter, we consider the optimal scheduling of power-intensive plants that can provide interruptible load. Again, we apply robust optimization to model the uncertainty in reserve demand; however, unlike in Chapter 5, we incorporate recourse decisions in a multistage framework by applying linear decision rules.

This chapter is organized as follows. Section 6.1 provides a brief literature review on interruptible load and adjustable robust optimization. Given the problem statement in Section 6.2, the uncertain scheduling model is presented in Section 6.3. The adjustable robust counterpart is developed in Section 6.4. In Section 6.5, the proposed model is applied to an illustrative example as well as an industrial air separation case. In Section 6.6, we close with a summary of the results.

6.1. Background

As mentioned in Section 1.3, one distinguishes between dispatchable and nondispatchable DR. Dispatchable DR refers to load adjustment capacities that consumers provide to the grid operator such that these capacities can be dispatched to maintain grid stability or in times of emergency. The grid operator has control over dispatchable DR resources by either direct load control or by sending load adjustment requests to the consumers. In nondispatchable DR, consumers are not obliged to
meet any load change requests by the grid operator, but rather choose to adjust their power consumption profiles based on price signals from the electricity market. Most existing works only consider nondispatchable DR; scheduling models involving dispatchable DR are very scarce, mainly due to the challenge of accounting for the inherent uncertainty in the problems.

Interruptible load is a dispatchable DR resource. The power grid is designed to match electricity supply and demand at all times. When real-time electricity supply falls below the demand, e.g. due to generator failures or sudden load changes, backup capacities are called upon in order to eliminate the supply-demand gap. One type of such backup capacity is called operating reserve, which has to be dispatched within minutes upon request. Providing reserve capacity is lucrative because the reserve provider is rewarded even when no actual dispatch is required. Operating reserve can be provided by power generation facilities that are able to quickly increase electricity supply. Alternatively, the supply-demand gap can be eliminated by reducing demand. Therefore, electricity consumers also have the opportunity to provide operating reserve if they possess the flexibility to quickly reduce their electricity consumption. Such operating reserve provided by electricity consumers is also referred to as interruptible load.

In the power systems literature, there is a large body of work on interruptible load management at the grid level. Some of the main questions addressed in these works are: How can interruptible load improve grid reliability and performance (Fotuhi-Firuzabad & Billinton, 2000; Bai et al., 2006; Aminifar et al., 2009)? Given offer curves from the interruptible load providers, how much interruptible load should the load serving entity procure (Tuan & Bhattacharya, 2003; Hatami et al., 2009a)? For an efficient reserve market, how should interruptible load be priced (Aalami et al., 2010)? In all these models, very simplistic representations of the electricity consumers are applied. However, in order to answer the fundamental question of how much interruptible load a consumer is really able and willing to provide, a consumer’s perspective has to be taken, and more detailed models have to be used.

The financial incentive for providing operating reserve reflects the value of flexible resources that can react quickly to unexpected changes in the power grid. The inherent uncertainty here is that one does not know in advance when and how much reserve will be needed. In Chapter 5, we have considered providing operating reserve by using the power generation capability of a cryogenic energy storage system. To account for the uncertainty in reserve demand, a robust optimization
6. Adjustable Robust Provision of Interruptible Load

Approach has been applied, which also provides the flexibility of adjusting the level of conservatism in the solution. Similarly, Vujanic et al. (2012) apply robust optimization to consider the scheduling of a batch plant that provides interruptible load. Here, the uncertainty in the time of required reserve dispatch is assumed to affect the start times of the scheduled tasks. However, in the model proposed by Vujanic et al. (2012), it is assumed that the amount of interruptible load provided is known, i.e., it is not a decision variable.

In robust optimization (Ben-Tal et al., 2009), the worst case is optimized while guaranteeing feasibility for all possible realizations of the uncertainty, which is described by an uncertainty set. When providing operating reserve, dispatch upon request has to be guaranteed since otherwise, one has to pay very high penalties, or may not even be allowed to participate in the reserve market. Hence, robust optimization is a natural choice for solving problems involving operating reserve. However, a major drawback of the traditional robust optimization approach—as applied in Chapter 5 and by Vujanic et al. (2012)—is that it does not account for recourse (reactive actions after the realization of the uncertainty); hence, the solution may be overly conservative. To overcome this limitation, the concept of adjustable robust optimization has been developed in recent years (Ben-Tal et al., 2004; Kuhn et al., 2011). The main idea is to include recourse in the form of decision rules that are functions of the uncertain parameters. If tractable (typically linear) decision rules are chosen, a robust counterpart formulation can be derived in the same fashion as in traditional static robust optimization. The adjustable robust optimization approach has been successfully applied to various operations research problems, such as inventory management (Ben-Tal et al., 2004), project management (Chen & Zhang, 2009a), and logistics planning (Ben-Tal et al., 2011).

In this work, we develop an MILP scheduling model for power-intensive continuous processes that participate in the reserve market by providing interruptible load. We apply an adjustable robust optimization approach to model the uncertainty in load reduction demand under a multistage decision-making setting, and derive a computationally tractable formulation for the adjustable robust model.

6.2. Problem Statement

Consider a power-intensive continuously operated plant that can produce a certain set of products, for which given demands have to be satisfied. Inventory capacities exist for storable products, and additional products can be purchased at given
costs. We assume that for fixed production, all production costs except for the cost of electricity are constant. In this way, for optimization purposes, the total operating cost only consists of the electricity cost and the cost of purchasing products. Electricity prices, which are time-sensitive, are assumed to be known for the scheduling horizon.

Besides selling products, the plant can gain additional revenue from providing operating reserve in the form of interruptible load, which is capacity for load reduction that the grid operator can request from the plant in case of contingency. Here, the load reduction is measured with respect to the plant’s target power consumption. The interruptible load provider is rewarded regardless how much load reduction is actually required, which is uncertain. Here, we assume that besides the corresponding reduction in electricity cost, no additional payment is made to the interruptible load provider when load is actually reduced upon request. Note that depending on the operating reserve market, this assumption may be relaxed.

The goal is to find a production schedule over a given time horizon that guarantees satisfaction of all product demand under every possible realization of the uncertainty, which lies in the actual demand for load reduction. Also, the solution is considered optimal if it minimizes net operating cost for the worst case, where the net operating cost is primarily the electricity cost and product purchase cost minus the revenue from providing interruptible load. In this problem, the here-and-now decisions are the modes of operation, the target production rates for each product, and the committed purchase amounts for each product in each time period of the scheduling horizon. The wait-and-see decisions are the changes in production rates and product purchases if load reduction is requested or has been requested in previous time periods.

6.3. Uncertain Scheduling Model

The proposed discrete-time MILP scheduling model is to a large extent a special case of the model developed in Chapter 3. Hence, we only provide brief descriptions of the scheduling constraints, and focus on the modeling of interruptible load and the derivation of the robust formulation in the next sections. Note that unless specified otherwise, all continuous variables presented in this section are constrained to be nonnegative.
6. ADJUSTABLE ROBUST PROVISION OF INTERRUPTIBLE LOAD

6.3.1. Production Scheduling Model

The CRS model for the production process is represented by the following set of constraints:

\[ PD_{it} = \sum_m \sum_{r \in R_m} PD_{mrit} \quad \forall \, i, t \in \mathcal{T} \]  

(6.1a)

\[ PD_{mrit} = \sum_{j \in J_{mr}} \lambda_{mrjt} v_{mrji} \quad \forall \, m, r \in R_m, i, t \in \mathcal{T} \]  

(6.1b)

\[ \sum_{j \in J_{mr}} \lambda_{mrjt} = \bar{y}_{mrt} \quad \forall \, m, r \in R_m, t \in \mathcal{T} \]  

(6.1c)

\[ EC_t = \sum_m \sum_{r \in R_m} \left( \delta_{mr} \bar{y}_{mrt} + \sum_i \gamma_{mri} PD_{mrit} \right) \quad \forall \, t \in \mathcal{T} \]  

(6.1d)

\[ y_{mt} = \sum_{r \in R_m} \bar{y}_{mrt} \quad \forall \, m, t \in \mathcal{T} \]  

(6.1e)

\[ \sum_{m} y_{mt} = 1 \quad \forall \, t \in \mathcal{T} \]  

(6.1f)

\[ y_{mt} \in \{0, 1\} \quad \forall \, m, t \in \mathcal{T} \]  

(6.1g)

\[ \bar{y}_{mrt} \in \{0, 1\} \quad \forall \, m, r \in R_m, t \in \mathcal{T} \]  

(6.1h)

where \( R_m \) is the set of subregions in mode \( m \), and \( J_{mr} \) is the set of vertices of subregion \( r \in R_m \). The binary variable \( y_{mt} \) equals 1 if mode \( m \) is selected in time period \( t \), whereas \( \bar{y}_{mrt} \) equals 1 if subregion \( r \in R_m \) is selected in time period \( t \). The amount of product \( i \) produced in time period \( t \) is denoted by \( PD_{it} \). Associated with \( PD_{it} \) is the disaggregated variable \( PD_{mrit} \) for subregion \( r \in R_m \), which is expressed as a convex combination of the corresponding vertices, \( v_{mrji} \). The amount of electricity consumed, \( EC_t \), is a linear function of \( PD_{it} \) with a constant \( \delta_{mr} \) and coefficients \( \gamma_{mri} \) specific to the selected subregion.

The transition constraints are the same as the ones presented in Chapter 4, but are shown here again for the sake of completeness:

\[ \sum_{m' \in TR_m} z_{m',t-1} - \sum_{m' \in TR_m} z_{mm',t-1} = y_{mt} - y_{m,t-1} \quad \forall \, m, t \in \mathcal{T} \]  

(6.2a)

\[ y_{mt} \bar{g}_{m't} \geq \sum_{k=1} \sum_{m''} z_{mm',t-k} \quad \forall \, (m, m') \in TR, t \in \mathcal{T} \]  

(6.2b)

\[ z_{mm',t-\bar{g}_{m'm''}} = z_{m'm''} \quad \forall \, (m, m', m'') \in SQ, t \in \mathcal{T} \]  

(6.2c)

\[ z_{mm't} \in \{0, 1\} \quad \forall \, (m, m') \in TR, t \in \mathcal{T} \]  

(6.2d)
where $\overline{TR}_m = \{ m' : (m', m) \in TR \}$ and $\overline{TR}_m = \{ m' : (m, m') \in TR \}$ with $TR$ being the set of all possible mode-to-mode transitions, $SQ$ is the set of predefined sequences, $\theta_{mm'}$ is the minimum stay time in mode $m'$ after switching to it from mode $m$, and $\bar{\theta}_{mm'm''}$ is the fixed stay time in mode $m'$ in the corresponding sequence. The binary variable $z_{mm't}$ equals 1 if and only if the plant switches from mode $m$ to mode $m'$ at time $t$.

The mass balance constraints are stated in the following:

\begin{align*}
IV_{it} &= IV_{i,t-1} + PD_{it} - SL_{it} \quad \forall \ i, t \in T \\
IV_{it}^{\text{min}} &\leq IV_{it} \leq IV_{it}^{\text{max}} \quad \forall \ i, t \in T \\
SL_{it} + PC_{it} &= D_{it} \quad \forall \ i, t \in T
\end{align*}

where the inventory level for product $i$ at time $t$ is denoted by $IV_{it}$, $SL_{it}$ is the amount sold, and $PC_{it}$ is the amount purchased from other sources.

The required initial conditions are given in the following:

\begin{align*}
IV_{i,0} &= IV_{i}^{\text{ini}} \quad \forall \ i \\
y_{m,0} &= y_{m}^{\text{ini}} \quad \forall \ m \\
z_{mm't} &= z_{mm't}^{\text{ini}} \quad \forall (m, m') \in TR, -\theta_{mm'}^{\text{max}} + 1 \leq t \leq -1
\end{align*}

with $\theta_{mm'}^{\text{max}} = \max \left( \max_{(m, m') \in TR} \{ \theta_{mm'} \}, \max_{(m, m', m'') \in SQ} \{ \bar{\theta}_{mm'm''} \} \right)$.

### 6.3.2. Interruptible Load Constraints

Interruptible load can be seen as the capability of a plant to reduce its electricity load within a short period of time. It can hence be used as an operating reserve resource to release the stress on the power grid in times of contingency. When interruptible load is provided, the plant still operates at its planned target production level, but has to be ready to respond to load reduction requests. When such a request actually occurs, the plant has to deviate from its target production rate such that the requested load reduction is achieved.

To model the provision of interruptible load, we first replace $PD_{mrit}$ by the following sum:

\begin{equation}
PD_{mrit} = PD_{mrit} + PD_{mrit} \quad \forall \ m, r \in R_m, i, t \in T
\end{equation}
6. Adjustable Robust Provision of Interruptible Load

where $\overline{PD}_{mrit}$ is the target production rate and $\overline{PD}_{mrit}$ is the response decrease in production rate when load reduction is required, in which case $\overline{PD}_{mrit}$ takes a negative value. The reduction in power consumption associated with the decrease in production with respect to the target production rate has to be at least the amount of requested load reduction, $LR_t$, as stated in the following constraint:

$$\sum_{m \in R_m} \sum_{i} \gamma_{mri} \overline{PD}_{mrit} \leq -LR_t \quad \forall \ t \in T$$

(6.6a)

$$x_t \in \{0, 1\} \quad \forall \ t \in T$$

(6.6b)

where $LR_t$ is an uncertain parameter whose characteristics will be discussed in detail later.

We further define a binary variable $x_t$, which equals 1 if interruptible load is provided in time period $t$. When interruptible load is provided, there may be lower and upper bounds on the provided amount as stated in the following:

$$IL_t^{\text{min}} x_t \leq IL_t \leq IL_t^{\text{max}} x_t \quad \forall \ t \in T$$

(6.7)

where $IL_t$ is the amount of interruptible load provided in time period $t$.

6.3.3. Objective Function

The objective is to minimize the total net operating cost, $TC$, which is defined as the sum of the electricity cost and the product purchase cost minus the revenue gained from providing interruptible load, as stated in the following equation:

$$TC = \sum_{t \in T} \left( \alpha_t^{EC} EC_t + \sum_i \alpha_{it}^{PC} PC_{it} - \alpha_t^{IL} IL_t \right)$$

(6.8)

where $\alpha_t^{EC}$, $\alpha_t^{PC}$, and $\alpha_t^{IL}$ are price coefficients.

6.3.4. Uncertain Optimization Problem

The uncertainty in the model lies in the parameter $LR$ since one does not know in advance when and how much load reduction will be required. However, when a plant provides interruptible load, it has to guarantee that load reduction up to the committed amount can be achieved when it is requested; noncompliance would result in very high penalties, or one may not even be allowed to participate in the market. Hence, in this uncertain optimization problem, we seek a robust solution
in a sense that it has to be feasible for every possible realization of the uncertain parameter.

Possible realizations of the uncertainty are defined in terms of an uncertainty set $U$. Here, the required load reduction, $LR$, cannot be higher than the amount of provided interruptible load, $IL$. Thus, the uncertainty set depends on $IL$, i.e. $U = U(IL)$; hence, this is a case of endogenous uncertainty as the uncertainty depends on the decisions made.

The uncertain optimization problem is formulated as follows:

$$\min \max_{LR \in U(IL)} \{TC\} \quad (6.9)$$

s.t. Eqs. (6.1)–(6.8) $\forall LR \in U(IL)$

where $LR = [LR_1, LR_2, \ldots, LR_t]^T$ and $IL = [IL_1, IL_2, \ldots, IL_t]^T$. It is stated that all constraints have to be feasible for any possible realization of $LR$ with respect to $U(IL)$. The objective is to minimize $\max_{LR \in U(IL)} \{TC\}$, which is the highest possible total net operating cost. In other words, the optimal objective function value corresponds to the worst case, which provides an upper bound on the total net operating cost for every realization of the uncertainty.

### 6.3.5 Uncertainty Set

Given a committed amount of interruptible load in time period $t$, $IL_t$, $LR_t$ can only take values between zero and $IL_t$. The uncertainty set can simply be formulated as

$$U(IL) = \{ LR : 0 \leq LR_t \leq IL_t \ \forall \ t \in T \} \quad (6.10)$$

which is obviously very conservative since $LR_t$ could take the value of $IL_t$ for all $t$. In order to reduce the level of conservatism, we apply the “budget of uncertainty” approach (Bertsimas & Sim, 2004) and adopt the uncertainty set proposed in Chapter 5:

$$W(IL) = \left\{ w : \left( LR_t = IL_t w_t, \ 0 \leq w_t \leq 1 \ \forall \ k \in K_{1,t}, \ \sum_{k=1}^{t} w_k \leq \Gamma_t \right) \ \forall \ t \in T \right\} \quad (6.11)$$

where in general, $K_{tt'} = \{ t, t+1, \ldots, t' \}$, $w_t = LR_t / IL_t$ is the normalized required load reduction, and $\Gamma_t$ is a budget parameter limiting the cumulative load reduction required up to time $t$. By changing $\Gamma_t$, the level of conservatism can be ad-
justed. In practice, the budget parameters can be chosen based on historical data; alternatively, depending on the market, there may be a strict limit on the number of times in which load reduction can be requested during a specific time horizon, which can be used to set $\Gamma_t$. Note that in order to have the desired effect, $\Gamma_t$ has to be monotonically increasing with $t$.

6.4. Adjustable Robust Counterpart

The uncertain optimization problem given by Eqs. (6.9) is a semi-infinite program that cannot be readily solved. Hence, we transform it into a problem with a finite number of constraints; in particular, the new formulation guarantees feasibility for every realization of the uncertainty with respect to the uncertainty set, and is therefore referred to as the robust counterpart. Here, we apply an adjustable robust optimization approach (Ben-Tal et al., 2004) to account for recourse.

In general, wait-and-see decisions can be expressed as functions of the uncertainty; hence, we have:

$$PD_{mrit} = PD_{mrit} + PD_{mrit}(w) \quad \forall \ m, r \in R_m, i, t \in T$$

$$PC_{it} = PC_{it} + PC_{it}(w) \quad \forall \ i, t \in T$$

where $PD_{mrit}$ and $PC_{it}$ are the target production rate and the committed product purchase, respectively, which are here-and-now decisions. The change in production rate depending on the realization of the uncertainty $w$ is denoted by $PD_{mrit}$; it includes the decrease in production rate when load reduction is requested as well as the possible increase in production rate after load reduction in order to make up for the loss in production. Similarly, $PC_{it}$ is the increase in product purchase as a function of $w$.

6.4.1. Linear Decision Rules

Generally, the functions of $w$ in Eqs. (6.12) could take any form. However, it is computationally intractable to optimize over the infinitely large set of all possible functions. Hence, in order to obtain a tractable formulation, we restrict ourselves to linear functions with respect to $w$, also referred to as linear decision rules, which are as follows:

$$PD_{mrit}(w) = \sum_{k=t-\zeta}^{t} p_{mritk} w_k \quad \forall \ m, r \in R_m, i, t \in T$$

$$PC_{it}(w) = \sum_{k=t}^{T} p_{itk} w_k \quad \forall \ i, t \in T$$
where $p_{mrkt}$ and $q_{itk}$ are variables that define the linear decision rules. This formulation allows multistage decision-making since at each time period $t$, the recourse decision depends on the uncertain parameters that have been realized in the preceding $\zeta_t$ time periods as well as the current time period, i.e. $w_k$ for $k \in K_{t-\zeta_t}$. The parameter $\zeta_t$ can be any integer between zero and $t-1$; it can be seen as another parameter for adjusting the level of conservatism. The greater $\zeta_t$ is, the more realized uncertainty is considered in the recourse; hence, an improved solution may be obtained. However, the problem size increases with $\zeta_t$. Therefore, in large-scale problems, a $\zeta_t < t-1$ can be chosen in order to achieve a trade-off between level of conservatism and problem size.

Eq. (6.13c) states that $p_{mrkt}$ for $k = t$ has to be nonpositive since in case load reduction is requested in time period $t$, the plant has to react with load reduction. However, as expressed in Eq. (6.13d), $p_{mrkt}$ is nonnegative for $k < t$ because if load was reduced in time period $k$, the natural recourse action in time period $t > k$ is an increase in production rate to make up for the loss in production. In the case of product purchase, only positive recourse makes sense; hence, $q_{itk}$ is restricted to be nonnegative.

### 6.4.2. Reformulation of the Plant Model

In order to construct the robust counterpart, we have to reformulate the CRS models that characterize the plant such that each polyhedral subregion is described by a set of inequalities instead of equalities. There are two commonly used representations of a polytope: the V-representation and the H-representation. In the V-representation, a point in the polytope is expressed as a convex combination of the polytope’s vertices; we have applied this representation in the CRS models through Eqs. (6.1b)–(6.1c). In the H-representation, the polytope is represented by a set of linear inequalities, where each inequality corresponds to a supporting hyperplane. The V- and H-representations are illustrated in the example shown in Figure 6.1.

We replace Eqs. (6.1b)–(6.1c) by the following constraints which constitute the
6. Adjustable Robust Provision of Interruptible Load

Figure 6.1: The V-representation makes use of the polytope’s vertices; the H-representation is formed by the supporting hyperplanes.

H-representations of each subregion in the CRS models:

\[
\begin{align*}
\sum t a_{mrfi} P D_{mrt}(w) & \geq b_{mrf} \bar{y}_{mrt} \quad \forall m, r \in R_m, f \in F_{mr}, t \in T \quad (6.14a) \\
0 & \leq P D_{mrt}(w) \leq P D_{mrt}^{\max} \quad \forall m, r \in R_m, i, t \in \mathcal{T} \quad (6.14b)
\end{align*}
\]

where \(a_{mrfi}\) and \(b_{mrf}\) are parameters defining the inequalities, and \(F_{mr}\) is the set of supporting hyperplanes associated with subregion \(r\) of mode \(m\).

6.4.3. Elimination of State Variables

Now we remove all “state variables”, i.e. variables that can be simply represented as linear functions of the decision variables. After eliminating the state variables, we obtain the following uncertain optimization problem:

\[
\begin{align*}
\min \max_{w \in W(IL)} & \left\{ \sum t \alpha_{EC} \sum m \sum r R_m \left[ \delta_{mr} \bar{y}_{mrt} + \gamma_{mrt} \left( P D_{mrt} + \sum t \delta_{k=t-\zeta_t} p_{mrtik} w_k \right) \right] \\
& + \sum t \sum i \alpha_{il}^{PC} \left( P C_{it} + \sum k=t-\zeta_t q_{itk} w_k \right) - \sum t \alpha_{il}^{IL} I L_t \\
& + \sum m \sum r R_m \sum t \delta_{mrt} \left( -p_{mrtik} - \sum t \delta_{k=t-\zeta_t} p_{mrtik} \right) + \sum i \sum t \delta_{il}^{IL} I L_t \right\} \\
\text{s.t.} & \quad \text{Eqs. (6.1e)–(6.1f), (6.2), (6.4b)–(6.4c), (6.7), (6.13c)–(6.13e)} \\
& \quad \sum t a_{mrfi} \left( P D_{mrt} + \sum t \delta_{k=t-\zeta_t} p_{mrtik} w_k \right) \geq b_{mrf} \bar{y}_{mrt} \quad \forall m, r \in R_m, f \in F_{mr}, t \in \mathcal{T} \quad (6.15b)
\end{align*}
\]
6. Adjustable Robust Provision of Interruptible Load

\[ 0 \leq PD_{mrt} + \sum_{k=t-\zeta_t}^{t} p_{mrik} w_k \leq PD_{mrt}^\text{max} \hat{y}_{mrt} \quad \forall \, m, r \in R_m, i, t \in T \quad (6.15c) \]

\[ IV_t^\text{min} \leq IV_{i,0} + \sum_{k=1}^{t} \left( \sum_{m \in R_m} \sum_{r \in R_m} \left( PD_{mrt} + \sum_{l=k-\zeta_k}^{k} p_{mrlk} w_l \right) \right) + \left( PC_{ik} + \sum_{l=k-\zeta_k}^{k} q_{ikl} w_l \right) - D_{ik} \leq IV_t^\text{max} \quad \forall \, i, t \in T \quad (6.15d) \]

\[ \sum_{m \in R_m} \sum_{r \in R_m} \sum_{i \in R_i} \gamma_{mri} \sum_{k=t-\zeta_t}^{t} p_{mrik} w_k \leq -IL_t w_t + \Omega (1 - x_t) \quad \forall \, t \in T \quad (6.15e) \]

\[ PD_{mrt} \geq 0 \quad \forall \, m, r \in R_m, i, t \in T \quad (6.15f) \]

\[ PC_{it} \geq 0 \quad \forall \, i, t \in T \quad (6.15g) \]

\[ x_t \in \{0, 1\} \quad \forall \, t \in T \quad (6.15h) \]

\[ y_{mt} \in \{0, 1\} \quad \forall \, m, t \in T \quad (6.15i) \]

\[ \hat{y}_{mrt} \in \{0, 1\} \quad \forall \, m, r \in R_m, t \in T \quad (6.15j) \]

\[ z_{mm't} \in \{0, 1\} \quad \forall \, (m, m') \in TR, t \in T \quad (6.15k) \]

\forall \, w \in W(\text{IL})

where costs for recourse actions have been added to the objective function. The last line states that all equations have to simultaneously hold for all \( w \in W(\text{IL}) \). Note that the smallest possible value for the big-M parameter \( \Omega \) is \( \max_{m, r \in R_m} \Delta EC_{mrt}^{\text{max}} \) with

\[ \Delta EC_{mrt}^{\text{max}} = \max_i \gamma_{mri} (PD_{mri}^{\text{up}} - PD_{mri}^{\text{lo}}) \]

s.t. \[ \sum_i a_{mrfi} PD_{mri}^{\text{up}} \geq b_{mrf} \quad \forall \, f \in F_{mr} \quad (6.16) \]

\[ \sum_i a_{mrfi} PD_{mri}^{\text{lo}} \geq b_{mrf} \quad \forall \, f \in F_{mr} \]

where \( \Delta EC_{mrt}^{\text{max}} \) is the maximum load change that can be achieved in subregion \( r \) of mode \( m \).

### 6.4.4. Linearly Adjustable Robust Counterpart

By using techniques commonly applied in robust optimization, the uncertain optimization problem with recourse given by Eqs. (6.15) can be reformulated into a finite-dimensional problem. The resulting linearly adjustable robust counterpart (ARC), for which the detailed derivation is presented in Appendix C, is shown in
6. Adjustable Robust Provision of Interruptible Load

the following:

\[
\begin{align*}
\text{min} & \quad \sum_{i \in T} \left[ \sum_{m \in R_m} \sum_{r \in R_m} \alpha_{mr}^{EC} \left( \delta_{mr} \bar{y}_{mrt} + \sum_{i} \gamma_{mrt} \bar{PD}_{mrt} \right) + \sum_{i} \alpha_{it}^{PC} \bar{PC}_{it} - \alpha_{it}^{IL} L_{it} \right] \\
& \quad + \sum_{m \in R_m} \sum_{r \in R_m} \sum_{k} \alpha_{mr}^{RP} \left( -p_{mr}^{it} + \sum_{i} p_{mrt} \right) + \sum_{i} \sum_{k} \alpha_{it}^{RQ} q_{itk} \\
& \quad + \left( \Gamma_t u^A + \sum_{k=1}^{t} s_k^{A} \right) 
\end{align*}
\]

\text{s.t.} \quad \text{Eqs. (6.1e)} - (6.1f), (6.2), (6.4b) - (6.4c), (6.7), (6.13c) - (6.13e), (6.15f) - (6.15k)

\[
\begin{align*}
\sum_{i} a_{mrf} \bar{PD}_{mrit} + \left( \Gamma_t u^B_{mrf} + \sum_{k=t-\zeta_i}^{t} s_{mrf}^{B} \right) & \geq b_{mrf} \bar{y}_{mrt} \\
& \quad \forall m, r \in R_m, f \in F_m, t \in T \quad (6.17b) \\
\bar{PD}_{mrit} + \left( \Gamma_t u^C_{mrit} + \sum_{k=t-\zeta_i}^{t} s_{mrit}^{C} \right) & \leq \bar{PD}_{mri}^{max} \bar{y}_{mrt} \\
& \quad \forall m, r \in R_m, i, t \in T \quad (6.17c) \\
\bar{PD}_{mrit} + \left( \Gamma_t u^D_{mrit} + \sum_{k=t-\zeta_i}^{t} s_{mrit}^{D} \right) & \geq 0 \\
& \quad \forall m, r \in R_m, i, t \in T \quad (6.17d) \\
IV_{i,0} + \sum_{k=1}^{t} \left( \sum_{m \in R_m} \sum_{r \in R_m} \bar{PD}_{mrik} + \bar{PC}_{ik} - D_{ik} \right) \\
& \quad + \left( \Gamma_t u^E_{it} + \sum_{k=1}^{t} s_{itk}^{E} \right) \leq IV_{i}^{max} \quad \forall i, t \in T \quad (6.17e) \\
IV_{i,0} + \sum_{k=1}^{t} \left( \sum_{m \in R_m} \sum_{r \in R_m} \bar{PD}_{mrik} + \bar{PC}_{ik} - D_{ik} \right) \\
& \quad + \left( \Gamma_t u^F_{it} + \sum_{k=1}^{t} s_{itk}^{F} \right) \geq IV_{i}^{min} \quad \forall i, t \in T \quad (6.17f) \\
\Gamma_t u^G_{it} + \sum_{k=t-\zeta_i}^{t} s_{itk}^{G} & \leq \Omega(1 - x_i) \quad \forall t \in T \quad (6.17g) \\
u^A + s_k^{A} & \geq \sum_{i} \left( \sum_{m \in R_m} \sum_{r \in R_m} \alpha_{mr}^{EC} \gamma_{mrt} p_{mrt} + \alpha_{it}^{PC} q_{itk} \right) \\
& \quad \forall k \in K_{1,t} \quad (6.17h) \\
u^A & \geq 0 \quad (6.17i) \\
s_k^{A} & \geq 0 \quad \forall 1 \leq k \leq t \quad (6.17j) \\
u^B_{mrf} + s_{mrf}^{B} & \leq \sum_{i} a_{mrf} p_{mrit} \quad \forall m, r \in R_m, f \in F_m, t \in T, k \in K_{t-\zeta_i,t} \quad (6.17k) \\
u^B_{mrf} & \leq 0 \quad \forall m, r \in R_m, f \in F_m, t \in T \quad (6.17l) \\
\sum_{i} a_{mrf} p_{mrit} & \leq 0 \quad \forall m, r \in R_m, f \in F_m, t \in T, k \in K_{t-\zeta_i,t} \quad (6.17m) \\
u^C_{mrit} + s_{mrit}^{C} & \geq p_{mrit} \quad \forall m, r \in R_m, i, t \in T, k \in K_{t-\zeta_i,t} \quad (6.17n) \\
\end{align*}
\]
6. Adjustable Robust Provision of Interruptible Load

\begin{align}
  u_{m rit}^C & \geq 0 \quad \forall m, r \in R_m, i, t \in T \\
  s_{m ritk}^C & \geq 0 \quad \forall m, r \in R_m, i, t \in T, k \in K_{t-\zeta t} \\
  u_{m rit}^D + s_{m ritk}^D & \leq p_{m ritk} \quad \forall m, r \in R_m, i, t \in T, k \in K_{t-\zeta t} \\
  u_{m rit}^D & \leq 0 \quad \forall m, r \in R_m, i, t \in T \\
  s_{m ritk}^D & \leq 0 \quad \forall m, r \in R_m, i, t \in T, k \in K_{t-\zeta t} \\
  u_{i t}^E + s_{i tk}^E & \geq \sum_{l=k}^{k+\zeta} \left( \sum_{m \in R_m} \sum_{r \in R_m} p_{m ritk} + q_{i l k} \right) \quad \forall i, t \in T, k \in K_{t-\zeta t} \\
  u_{i t}^E & \geq 0 \quad \forall i, t \in T \\
  s_{i tk}^E & \geq 0 \quad \forall i, t \in T, k \in K_{t-\zeta t} \\
  u_{i t}^F + s_{i tk}^F & \leq \sum_{l=k}^{k+\zeta} \left( \sum_{m \in R_m} \sum_{r \in R_m} p_{m ritk} + q_{i l k} \right) \quad \forall i, t \in T, k \in K_{t-\zeta t} \\
  u_{i t}^F & \leq 0 \quad \forall i, t \in T \\
  s_{i tk}^F & \leq 0 \quad \forall i, t \in T, k \in K_{t-\zeta t} \\
  u_{i t}^G + s_{i tk}^G & \geq \sum_{m \in R_m} \sum_{r \in R_m} \sum_{i \in I} \gamma_{m ri} p_{m ritk} \quad \forall t \in T, k \in K_{t-\zeta t} \\
  u_{i t}^G + s_{i tk}^G & \geq \sum_{m \in R_m} \sum_{r \in R_m} \sum_{i \in I} \gamma_{m ri} p_{m ritk} + IL_t \quad \forall t \in T \\
  u_{i t}^G & \geq 0 \quad \forall t \in T \\
  s_{i tk}^G & \geq 0 \quad \forall t \in T, k \in K_{t-\zeta t}
\end{align}

6.4.5. Remark on Robust Formulation Without Recourse

Usually, the adjustable robust optimization approach is compared to the traditional static robust optimization approach, which does not consider recourse. However, it turns out that formulating a robust counterpart without recourse for this problem is not generally possible without making further restrictive assumptions.

In order to account for uncertainty without the notion of recourse, the effect of the uncertainty on the model has to be exactly known. Typically, for tractability reasons, one has to be able to express this effect as a linear function of the uncertain parameters, which in our case are the load reduction demands in each time period. The necessary response to a load reduction request is a corresponding decrease in production, which is stated in Eq. (6.6). However, this response is not unique since with multiple products, there may be an infinite number of possible combinations of production rates that lead to the same amount of load reduction. Therefore, a robust model without recourse cannot be formulated unless very conservative assumptions are made, e.g. by fixing the production rates for all but one products a priori. For the limiting case of one single product, however, a traditional robust
optimization model without recourse can be formulated. The derivation of the corresponding robust counterpart is shown in Appendix D.

### 6.4.6. Tightening Constraints

In order to further improve the computational performance of the MILP model, redundant constraints are added that lead to a tighter formulation. First, we notice that the set of operating modes, \( M \), can be divided into two disjoint sets: a set of flexible modes and a set of inflexible modes, denoted by \( \overline{M} \) and \( \overline{M} \), respectively. In a flexible mode, the electricity consumption can be changed within a range; hence, it is suited for providing interruptible load. In contrast, an inflexible mode is represented by one single operating point and does not possess the flexibility for load adjustment. Inflexible modes are, for example, the off mode and modes representing transitional stages, such as startup and shutdown. With this insight, we know that all \( p_{mrtk} \) associated with inflexible modes have to be zero. Therefore, we can remove these \( p \)-variables as well as all corresponding constraints that result from the duals of the lower-level problems. By doing so, the numbers of variables and constraints can potentially be significantly reduced. The same insight can be used to formulate the following tightening constraints:

\[
\begin{align*}
x_t &\leq \sum_{m \in \overline{M}} y_{mt} \quad \forall t \in T \quad (6.18a) \\
x_t &\leq 1 - y_{mt} \quad \forall m \in \overline{M}, t \in T 
\end{align*}
\]

where Eq. (6.18a) states that \( x_t \) can only take the value 1 if a flexible mode is chosen in time period \( t \). Similarly, Eq. (6.18b) states that \( x_t \) has to take the value 0 if an inflexible mode is selected in time period \( t \).

### 6.5. Numerical Results

In the following, the proposed robust model is applied to an illustrative example as well as a real-world industrial air separation case. All models were implemented in GAMS 24.4.1 (GAMS Development Corporation, 2015a), and the commercial solver CPLEX 12.6.1 (IBM ILOG, 2015a) was applied to solve the MILPs on an Intel® Core™ i7-2600 machine at 3.40 GHz with 8 processors and 8 GB RAM running Windows 7 Professional.
6.5.1. Illustrative Example

In order to demonstrate the features of the model, we first apply it to a small illustrative example in which a single-product plant is considered. The scheduling problem is solved for a time horizon of 48 h while applying an hourly time discretization.

The given plant can operate in three different operating modes: off, startup, and on, where each mode is represented by one single convex region. Table 6.1 shows the constraints for the H-representations of each mode as well as the corresponding fixed and unit electricity consumption parameters. Note that only the on mode has the flexibility for load changes. Table 6.2 shows the possible mode transitions and the respective minimum stay times. Furthermore, in the predefined sequence off → startup → on, the plant has to remain in the startup mode for exactly 4 h.

Table 6.1: H-representations, fixed (δₘ) and unit (γₘ) electricity consumption for each operating mode (product and subregion indices have been omitted).

<table>
<thead>
<tr>
<th>Operating Mode</th>
<th>H-Representation</th>
<th>δₘ [kWh]</th>
<th>γₘ [kWh/kg]</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>0 ≤ PDₘₜ ≤ 0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>startup</td>
<td>5 ≤ PDₘₜ ≤ 5</td>
<td>0</td>
<td>60</td>
</tr>
<tr>
<td>on</td>
<td>80 ≤ PDₘₜ ≤ 160</td>
<td>1500</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 6.2: Possible transitions between the different operating modes and the corresponding minimum stay times.

<table>
<thead>
<tr>
<th>Transition from Mode m to Mode m'</th>
<th>Minimum Stay Time in Mode m' [h]</th>
</tr>
</thead>
<tbody>
<tr>
<td>off → startup</td>
<td>4</td>
</tr>
<tr>
<td>startup → on</td>
<td>6</td>
</tr>
<tr>
<td>on → off</td>
<td>8</td>
</tr>
</tbody>
</table>

At the beginning of the scheduling horizon, the plant is operating in the on mode, and it is assumed that no mode switching has occurred in the 8 time periods before the beginning of the scheduling horizon. The initial inventory is 1000 kg. The minimum and maximum inventory levels are set to 0 and 5000 kg, respectively, for all time points. The only exception is the end of the scheduling horizon at which the amount of product in inventory is constrained to be at least 1000 kg.

When interruptible load is provided in a time period, the provided amount has to be between 200 and 1600 kWh. The electricity and interruptible load prices are
shown in Figure 6.2. The price for purchasing additional products is $3/kg, and the unit costs for recourse, \( \alpha^{RP} \) and \( \alpha^{RQ} \), are set to $0.01/kg for all time periods.

![Figure 6.2: Electricity and interruptible load prices for the illustrative example.](image1)

Figure 6.2: Electricity and interruptible load prices for the illustrative example.

Figure 6.3 shows the results of the base case, in which the provision of interruptible load is not considered. Note that in the figure, the y-axes for the inventory profile and the product flows are shown on the left and right hand side, respectively. Positive columns indicate accumulation of product, while negative columns (demand) indicate depletion of product. The level of plant utilization, which is defined as the ratio between the amount of product produced and the maximum amount that can be produced, is at approximately 77\%, with the plant’s maximum production capacity being 160 kg/h. One can see that the plant is shut down for several hours (24–33 h) while still satisfying product demand. A small amount of product has to be purchased, yet the cost is greatly outweighed by the savings in electricity cost made possible by the temporary shutdown of the plant. The inventory level reaches its allowed minimum at the end of the scheduling horizon. The total cost for this optimal solution is $5556.

![Figure 6.3: Product flows and resulting inventory profile for the case without interruptible load.](image2)
6. Adjustable Robust Provision of Interruptible Load

In the following, the effect of interruptible load is evaluated for three types of scenarios, each varying one model parameter:

1. The possible extent of recourse decisions is varied by adjusting $\zeta_t$. We introduce an auxiliary parameter $\bar{\zeta}$, which denotes the maximum number of previously realized uncertain parameters that are considered in the decision rules, and set $\zeta_t$ such that $\zeta_t = \min\{\bar{\zeta}, t-1\}$. The level of conservatism decreases, but the model size increases with $\bar{\zeta}$.

2. The level of uncertainty depends on the definition of the uncertainty set. It is varied by adjusting the budget parameters $\Gamma_t$.

3. The level of plant utilization is varied by changing the demand.

Scenario Type 1: Varying Extent of Recourse

In order to show the value of recourse, different instances are created by varying $\bar{\zeta}$. The flexibility in the recourse actions increases with $\bar{\zeta}$ since more uncertain parameters from preceding time periods are taken into account and more terms appear in the linear decision rules. For all cases of scenario type 1, the level of uncertainty is the same; here, the budget parameter $\Gamma_t$ increases every 8 time periods by 1, i.e. maximum load reduction can only be requested once during the first 8 h, twice during the first 16 h, three times during the first 24 h, etc., and at most six times during all 48 h.

For the case of $\bar{\zeta} = 0$, i.e. the recourse decision in time period $t$ only depends on the uncertainty revealed in the same time period, the results are shown in Figures 6.4 and 6.5. Along with the electricity and interruptible load prices, Figure 6.4 shows the target load profile for the plant as well as the amount of interruptible load provided, which obviously has to be less than the target electricity consumption. Unlike in the case without interruptible load (c.f. Figure 6.3), the plant is not shut down in the middle but toward the end of the scheduling horizon, allowing the provision of large amount of interruptible load when the price is high.

Figure 6.5 shows the inventory profile and the corresponding product flows for the case of $\bar{\zeta} = 0$. In addition to the target production and purchase, the recourse actions in terms of reducing production (negative) and increasing purchase (positive) are shown. Negative production recourse indicates time periods in which interruptible load is provided. Since $\bar{\zeta} = 0$, when load reduction is requested, the only way to regain lost production is through additional product purchase. However, since purchasing products is expensive, not all lost production is made up in
Figure 6.4: Target electricity consumption profile and provided interruptible load for the case of \( \bar{\zeta} = 0 \), and price profiles.

the optimal solution. Instead, the solution suggests to overproduce such that an inventory buffer is created, which guarantees that all demand can be satisfied in every possible load reduction scenario. This inventory buffer is indicated by the final inventory level that is higher than its allowed minimum of 1000 kg. The resulting worst-case total cost is $5540, which is an almost negligible cost reduction of 0.3% compared to the case without interruptible load.

Figure 6.5: Target and recourse product flows and target inventory profile for the case of \( \bar{\zeta} = 0 \).

To improve the solution while robustifying against the same level of uncertainty, \( \bar{\zeta} \) is set to 47, which is the case with the highest possible recourse flexibility since at each time period, all previous uncertain parameters in the scheduling horizon are considered in the decision rule. Hence, recourse in the form of production increase can also be considered. Figure 6.6 shows the optimal solution for this case. Here, cumulative recourse actions are shown, i.e. for each time period \( t \), we plot
Several differences to the solution for the case of $\bar{\zeta} = 0$ can be observed:

- The plant is shut down during some high-price hours in the middle (24–33 h) rather than toward the end of the scheduling horizon.
- Less interruptible load is provided.
- Most lost production is made up by increasing production after the requested load reduction has occurred.
- No inventory buffer is created.

The total cost for this case is $5127, which is a cost reduction of 7.7% compared to the case without interruptible load. The cost savings are increased by more than 25 times compared to the case of $\bar{\zeta} = 0$, simply achieved by a more flexible and realistic modeling of the recourse.

Figure 6.6: Target and recourse product flows and target inventory profile for the case of $\bar{\zeta} = 47$.

Table 6.3 lists the cost values for various cases with different $\bar{\zeta}$. Note that the uncertainty-related worst-case cost—expressed in the term $\Gamma_t u^A + \sum_{k=1}^t s_k^A$ of the objective function—is not listed in the table because it is zero in all cases. One can see that the total cost decreases with $\bar{\zeta}$. Another observation is that the results for $\bar{\zeta} = 23$ and $\bar{\zeta} = 47$ are identical, which indicates that one may not require the model allowing maximum flexibility in the recourse to achieve the optimal solution. This is a useful insight since the model size increases with $\bar{\zeta}$; hence, there is a trade-off between level of conservatism and computational tractability. In this small example problem, the impact of increased model size is insignificant. However, as we
will show in the industrial case study in Section 6.5.2, it can cause severe deterioration of the computational performance.

Table 6.3: Costs and revenues in $ for cases with different $\bar{\zeta}$. Here, $C^{PD}$, $C^{PC}$, $R^{IL}$, $C^{RC}$, and $TC$ denote production cost, purchasing cost, revenue from providing interruptible load, recourse cost, and total cost, respectively.

<table>
<thead>
<tr>
<th>$\bar{\zeta}$</th>
<th>$C^{PD}$</th>
<th>$C^{PC}$</th>
<th>$R^{IL}$</th>
<th>$C^{RC}$</th>
<th>$TC$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7050</td>
<td>0</td>
<td>1536</td>
<td>26</td>
<td>5540</td>
</tr>
<tr>
<td>5</td>
<td>5024</td>
<td>900</td>
<td>554</td>
<td>13</td>
<td>5383</td>
</tr>
<tr>
<td>11</td>
<td>5087</td>
<td>789</td>
<td>703</td>
<td>17</td>
<td>5190</td>
</tr>
<tr>
<td>23</td>
<td>5087</td>
<td>789</td>
<td>768</td>
<td>19</td>
<td>5127</td>
</tr>
<tr>
<td>47</td>
<td>5087</td>
<td>789</td>
<td>768</td>
<td>19</td>
<td>5127</td>
</tr>
</tbody>
</table>

Scenario Type 2: Varying Level of Uncertainty

Having examined cases with moderate uncertainty, we now vary the level of uncertainty and in particular investigate the two extremes: the most uncertain case in which load reduction can be requested in all time periods ($\Gamma = [1, 2, \ldots, 48]^T$), and the least uncertain case in which load reduction request can only occur in one time period ($\Gamma = [1, 1, \ldots, 1]^T$). In both cases, $\bar{\zeta}$ is chosen to be 47.

As expected, the cost savings decrease with increasing level of uncertainty. In the most uncertain case, the total cost amounts to $5525, while it is only $4851 in the least uncertain case. It is worth taking a closer look at the results for the case with the highest level of uncertainty shown in Figure 6.7. Similar to the $\bar{\zeta} = 0$ case from Section 6.5.1.1 (c.f. Figure 6.5), the plant is only shut down toward the end of the scheduling horizon. However, significantly less interruptible load is provided. No inventory buffer is created; hence, since the solution has to be feasible for the particular scenario in which maximum load reduction is requested in all time periods, the total amount of regained production through recourse is the same as the total amount of interruptible load provided (in terms of decrease in production).

Scenario Type 3: Varying Plant Utilization

One main insight drawn from existing works on scheduling of power-intensive industrial processes is that the benefits of DR typically decrease with increasing level of plant utilization (Mitra et al., 2012a; Castro et al., 2013). The explanation
is that lower utilization implies higher process flexibility, which allows more effective load shifting. However, when interruptible load is provided, the relationship between plant utilization and cost savings is not that obvious. On the one hand, it is still true that flexibility decreases with plant utilization; on the other hand, higher utilization implies higher target production levels, which allow more interruptible load to be provided.

We apply the uncertainty set used in the type-1 scenario cases, set $\bar{\zeta} = 47$, and solve the problem for different levels of utilization. In the diagram shown in Figure 6.8, the absolute cost savings are plotted against the level of plant utilization. Before very high levels of utilization are reached, the general trend is that cost savings increase with plant utilization. However, the benefits decrease after a certain point because if the plant has to operate at almost full capacity to satisfy demand, recourse has to increasingly rely on additional product purchase, which is relatively expensive. In this case, the highest cost savings are achieved at a plant utilization of 95%.

### 6.5.2. Industrial Case Study

We now apply the proposed model to a real-world industrial case study provided by Praxair. Here, we consider an air separation plant that produces liquid oxygen (LO2) and liquid nitrogen (LN2). The scheduling horizon is one week, to which an hourly time discretization is applied resulting in 168 time periods. The PJM electricity market is considered, and the hourly day-ahead energy and operating reserve prices are taken from the week of June 23 to 29, 2014 (PJM Interconnection LLC, 2014c,a).
A base case is considered in which plant utilization is at 90%. For all instances, $\Gamma_t$ is chosen to increase by 1 every 24 time periods, i.e. load reduction can be requested up to 7 times over the whole week. This is a fairly conservative assumption since typically, dispatch of operating reserve is only requested a few times in months (EnerNOC, 2014). Figure 6.9 shows the LO2 and LN2 production, purchase, and inventory profiles for the case in which no interruptible load is provided. The optimal solution suggests to shift production as much as possible to times when the electricity price (shown in Figure 6.10) is low. The resulting total cost is 100 (normalized).

For the case when interruptible load is provided, various instances with different $\bar{\zeta}$ are created. For each instance, Table 6.4 lists the total costs, model sizes, and wall-clock computation times used to solve the MILPs to 0.1% optimality gap. As expected, the total cost decreases with increasing $\bar{\zeta}$ due to the higher flexibility in the recourse. However, the numbers of continuous variables and constraints grow with $\bar{\zeta}$, and lead to dramatic increases in computation time. Here, we see the clear trade-off between the level of conservatism in the model and its computational performance. For our further analysis, we choose $\bar{\zeta}$ to be 23, as the required computation time is reasonable for practical purposes and only minor improvement is achieved for $\bar{\zeta} > 23$.

Besides the electricity and interruptible load prices, Figure 6.10 shows the target load profile and the amount of interruptible load provided for the case of $\bar{\zeta} = 23$. It is worth pointing out that typically, as it is also the case here, high electricity prices coincide with high interruptible load prices. For those time periods, the
optimal solution reveals whether reducing production to save electricity cost or increasing production to provide more interruptible load is more beneficial. Figure 6.11 shows the target production, purchase, and inventory profiles as well as the recourse actions, which suggest to make up for the vast majority of the lost production by increasing production after load reduction. Compared to the base case without interruptible load, the total cost reduces by 1.8%, which may seem small but is actually significant at such industrial scale.

Figure 6.12 shows the relationship between cost savings and level of plant utilization. Similar to the illustrative example, the cost reduction compared to the solution without interruptible load reaches its maximum at approximately 95% plant utilization; it drops to zero at 100% since production at full capacity is needed to satisfy demand, and the cost of purchasing additional products as recourse outweighs the benefit of providing interruptible load.
Table 6.4: Total costs, model sizes, and computation times for instances with different $\tilde{\zeta}$.

<table>
<thead>
<tr>
<th>$\tilde{\zeta}$</th>
<th>$TC$</th>
<th># of Bin. Variables</th>
<th># of Cont. Variables</th>
<th># of Constraints</th>
<th>Wall-Clock Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>98.83</td>
<td>3282</td>
<td>82,670</td>
<td>84,604</td>
<td>185</td>
</tr>
<tr>
<td>5</td>
<td>98.48</td>
<td>3282</td>
<td>139,595</td>
<td>139,879</td>
<td>1056</td>
</tr>
<tr>
<td>11</td>
<td>98.33</td>
<td>3282</td>
<td>205,628</td>
<td>203,998</td>
<td>2598</td>
</tr>
<tr>
<td>17</td>
<td>98.29</td>
<td>3282</td>
<td>269,177</td>
<td>265,705</td>
<td>5846</td>
</tr>
<tr>
<td>23</td>
<td>98.20</td>
<td>3282</td>
<td>330,242</td>
<td>325,000</td>
<td>6476</td>
</tr>
<tr>
<td>35</td>
<td>98.16</td>
<td>3282</td>
<td>444,920</td>
<td>436,354</td>
<td>18,381</td>
</tr>
<tr>
<td>47</td>
<td>98.14</td>
<td>3282</td>
<td>549,662</td>
<td>538,060</td>
<td>23,280</td>
</tr>
</tbody>
</table>

Figure 6.10: Target electricity consumption profile and provided interruptible load for the case of $\tilde{\zeta} = 23$, and price profiles.

6.6. Summary

The work presented in this chapter has addressed the robust scheduling of power-intensive industrial processes that can participate in the operating reserve market by providing interruptible load. An adjustable robust optimization approach has been applied to formulate a discrete-time MILP model that accounts for the uncertainty in load reduction demand, while considering recourse actions in the form of linear decision rules. The solution of the proposed model is guaranteed to be robust with respect to a budget uncertainty set that can be adjusted to change the level of conservatism.

An illustrative example and a real-world industrial air separation case demonstrate the capability of the proposed model. The case studies show that significant financial benefits can be achieved by selling interruptible load. The results further
6. Adjustable Robust Provision of Interruptible Load

Figure 6.11: Target and recourse product flows and target inventory profile for the case of $\zeta = 23$.

Figure 6.12: Absolute cost savings vs. level of plant utilization in the industrial case study.
demonstrate the value of recourse as the cost savings increase with the extent of recourse, in this case namely the number of uncertain parameters in preceding time periods considered in the decision rules. However, this flexibility in the recourse comes at the cost of increased model size and computation time. This trade-off has to be carefully considered in large-scale problems.

Moreover, contrary to results in the literature indicating that DR is more effective in plants with lower utilization, we find that this is not true when interruptible load is provided. Here, the largest cost savings are achieved at a high, yet not maximum level of plant utilization. The explanation is that higher plant utilization allows larger amount of interruptible load to be provided, yet some flexibility is still required for the implementation of effective recourse.
After addressing industrial DSM at the plant level, we now take it to the supply chain level, where load shifting can be considered not only across time periods, but also across plants that may have different power contracts and serve different customers. In order to capture synergies in integrated supply chains, production and distribution operations have to be optimized simultaneously. In this chapter, we introduce the multiscale production routing problem (MPRP), which considers the coordination of production, inventory, distribution, and routing decisions in multicommodity supply chains with complex production facilities. We emphasize the complexity of the production processes because this distinguishes the MPRP from the classical production routing problem (PRP) that models the production side with a simple lot-sizing formulation, which cannot represent the operations of power-intensive processes with the desired accuracy. In order to solve large instances of the MPRP, we propose an iterative MILP-based heuristic approach that solves an integrated MILP model with a restricted set of candidate routes at each iteration and dynamically updates the set of candidate routes for the next iteration.

This chapter is organized as follows. Section 7.1 provides a literature review on production routing and solution methods that have been proposed for solving this problem. Given the problem statement in Section 7.2, an MILP formulation is proposed for the MPRP in Section 7.3. Section 7.4 provides a description of the proposed solution method. In Section 7.5, the proposed approach is applied to various MPRP instances, including an illustrative example to show the main features of the model, an extensive computational study to demonstrate the effectiveness of the solution method, and a real-world industrial-scale test case with data provided by Praxair. In Section 7.6, we close with a summary of the results.
7.1. Background

In today’s competitive market environment, it is becoming increasingly important for companies in the process industry to improve the performance of their supply chains. One widely acknowledged approach for achieving a more efficient and reliable supply chain is the integrated planning of multiple supply chain operations such as production, inventory, and distribution (Thomas & Griffin, 1996; Ercümc et al., 1999). Typically, these operations are optimized in a sequential manner. For example, one may first forecast the demand for each production plant and set up production plans that minimize production and inventory costs at the plants. Then, using the production decisions as inputs, distribution planning is performed, which minimizes the distribution costs. However, since the distribution planning is restricted by the production planning decisions, the solution may be suboptimal due to the lack of coordination. In contrast, with an integrated supply chain planning approach, several major planning decisions are optimized simultaneously, which can result in significant cost savings, as shown in some recent successful industrial implementations (Brown et al., 2001; Çetinkaya et al., 2009).

Among the integrated supply chain planning problems, the so-called production routing problem (PRP), also sometimes referred to as the production inventory distribution routing problem (PIDRP), is the most comprehensive one as it considers production, inventory, distribution, and routing decisions simultaneously. The PRP in its classical form can be formulated as an MILP and integrates two well-known problems, namely the lot-sizing problem (LSP) and the inventory routing problem (IRP), where the latter is an extension of the vehicle routing problem (VRP). For details on these more extensively studied subproblems of the PRP, we point the interested reader to the following references: Karimi et al. (2003) and Pochet & Wolsey (2006) for the LSP, Toth & Vigo (2002) and Laporte (2009) for the VRP, and Campbell et al. (1998) and Coelho et al. (2013) for the IRP. It is important to note that in the PRP, there is flexibility in the inventories not only at the production plants but also at the customer locations; hence, customer inventory levels are decision variables, and an IRP has to be solved as a subproblem. In many other integrated production and routing problems, fixed orders are assumed, which excludes the possibility of leveraging the customers’ storage capacities.

Although the PRP has received increased attention in recent years, the literature on this subject remains scarce. In one of the first works on production routing, Chandra & Fisher (1994) consider a single plant producing multiple products for
several customers. Two approaches are compared to each other, one in which the LSP and the IRP are solved sequentially, and another in which an integrated PRP is solved. Although the PRP is solved heuristically, the results show that cost reduction of 3–20% can be achieved by applying the integrated approach. Fumero & Vercellis (1999) solve a similar problem using Lagrangean relaxation and obtain cost savings of the same order of magnitude.

Due to its high combinatorial complexity, the PRP is notoriously hard to solve. As a result, most existing solution approaches involve various heuristic procedures. Like several others, Lei et al. (2006) propose a two-phase approach for solving the PRP. In Phase I, the integrated problem is solved, allowing only direct shipments from plants to customers. The resulting inefficiencies are handled in Phase II by minimizing the transportation cost for each plant and each time period with the plant-to-customer allocation decisions from Phase I. Bard & Nananukul (2009a, 2010) apply a branch-and-price algorithm in which column generation is applied to solve the LP relaxation at each node of the branch-and-bound tree. Here, each generated column corresponds a feasible routing schedule in a particular time period. Since the pricing subproblems are extensions of the VRP, they are very difficult to solve exactly; hence, a heuristic two-step algorithm is applied, which determines delivery quantities for each customer in each time period in the first step and then finds actual routes using a VRP tabu search code in the second step. Archetti et al. (2011) first solve an IRP heuristically while assuming infinite production at the plant. Then an LSP is solved with the production quantities obtained from solving the IRP. Finally, the solution is improved iteratively by removing and inserting two customers at a time. Absi et al. (2013) propose an iterative two-phase approach. The first phase involves solving a PRP in which the routing part is simplified by direct assignment of vehicles to customers and time periods with fixed visiting costs. Routes are then constructed in the second phase, and based on the routing solution, the visiting costs are updated for the next iteration.

Besides the general heuristics reviewed above, metaheuristics have also been applied to the PRP. Boudia et al. (2007) propose a greedy randomized adaptive search procedure (GRASP) consisting of two main phases: construction and local search. In the construction phase, an initial solution is generated by developing a production and distribution plan sequentially for each time period without creating inventory at the plant. Then in the local search phase, the routing plan is improved by simple local moves in the same time period as well as across multiple time periods. Boudia & Prins (2009) introduce a memetic algorithm, which can be
seen as a modified genetic algorithm that applies an improvement procedure to each generated solution. Bard & Nananukul (2009b) first solve an LSP with direct shipments and then apply tabu search to make routing decisions based on the solution of the LSP. Armentano et al. (2011) propose a similar tabu search algorithm and further incorporate a path relinking procedure, in which new solutions are generated by connecting high-quality solutions or solutions that exhibit contrasting features. Adulyasak et al. (2014b) apply an adaptive large neighborhood search (ALNS) algorithm, where at each iteration, a selection operator is applied to create a list of customer-time period combinations, and then a transformation operator is applied to remove or reinsert some of these customer-time period combinations to the current solution.

Only a handful of exact solution methods have been proposed for the PRP. Bard & Nananukul (2009a, 2010) propose a rigorous branch-and-price algorithm; however, it can only solve instances with up to 10 customers, 5 vehicles, and 2 time periods. Different branch-and-cut algorithms have been developed by Ruokokoski et al. (2010), Archetti et al. (2011), and Adulyasak et al. (2014a). In all proposed branch-and-cut procedures, subtour elimination constraints are added as cuts when solving the LP relaxations at the nodes of the branch-and-bound tree. In addition, Ruokokoski et al. (2010) formulate so-called generalized comb and 2-matching inequalities that can be used as cuts in the algorithm. Adulyasak et al. (2014a) apply branch-and-cut to two different formulations of the PRP, one with and one without a vehicle index. Computational tests show that the vehicle index formulation is superior in finding optimal solutions, whereas the nonvehicle index formulation generally provides better bounds on larger instances that cannot be solved to optimality. Adulyasak et al. (2015a) further apply the proposed branch-and-cut algorithm combined with Benders decomposition to solve the two-stage and multistage stochastic PRP with demand uncertainty.

For more details on formulations and solution algorithms for the classical PRP, we refer to the excellent review of Adulyasak et al. (2015b), from which the following insights, among other ones, can be drawn: (1) The vast majority of existing works only consider problems with one plant and one product. (2) The state-of-the-art heuristic algorithms for the PRP can solve instances with up to 200 customers and 20 time periods. (3) The best existing exact algorithms can handle single-vehicle instances with up to 80 customers and 8 time periods.

Our work is concerned with an extension of the classical PRP and is motivated by the challenge of managing production and distribution operations in industrial...
gas supply chains. In the so-called merchant liquid business, industrial gas companies distribute liquid products (liquid oxygen, nitrogen, argon, hydrogen, etc.) in bulk to the customers using tractor-trailers. The products can be stored in tanks at the customer sites. Here, the concept of vendor-managed inventory (VMI) is applied such that the industrial gas companies have control over their customers’ inventories. These are highly integrated supply chains with multiple products, multiple production plants, and typically hundreds of customers. Moreover, cryogenic air separation plants, which are used to produce high-quality industrial gases, are tightly integrated and highly power-intensive processes. Hence, when optimizing the operation of such plants, detailed scheduling models have to be applied that can capture all critical process features including time-sensitive prices, interdependent production rates, and constraints on transitions between operating points. This level of accuracy on the production side cannot be achieved by a lot-sizing model as used in the classical PRP formulation.

For the industrial gas supply chain case, Glankwamdee et al. (2008) formulate a simplified production-distribution LP model that does not consider routing decisions; instead, the distribution part is approximated by resource constraints on truck and driver hours required for the planned deliveries. Marchetti et al. (2014) propose a production routing framework in which a heuristic is applied to generate a number of routes a priori, where a route is defined as a set of customers that can be visited in one trip. These routes are then included in the integrated model such that the assignment of routes to available trucks can be optimized. A large-scale industrial test case with 2 products, 4 plants, 4 depots, 168 customers, and 14 time periods has been considered, for which CPLEX finds a good feasible solution with an optimality gap of 3.6% within 5 h. However, it should be mentioned that in this particular case, the delivery quantities are given, i.e. there is no inventory management at the customer sites involved. In their proposed frameworks, Glankwamdee et al. (2008) and Marchetti et al. (2014) apply rather simplistic models of the production processes, which can be a serious drawback as process dynamics are not accurately represented, and hence, solutions may be suboptimal or even infeasible when implemented in practice. Zamarripa et al. (2016) apply a rolling horizon heuristic to large-scale instances of the model proposed by Marchetti et al. (2014), obtaining near-optimal solutions in shorter computation times.

The goal of this work is to develop a production routing framework that can consider large-scale multicommodity multiplant supply chains with complex production facilities, such as industrial gas supply chains. The desired outputs are
7. Multiscale Production Routing in Power-intensive Supply Chains

twofold: a production schedule that can be readily implemented, and plant-to-
customer allocation decisions that can be used as input for a subsequent detailed
inventory routing tool, such as the one developed by Dong et al. (2014). We propose
a multiscale PRP (MPRP) model involving two time grids, a fine one for produc-
tion scheduling and a coarse one for distribution planning. A detailed production
scheduling model is applied to capture all critical operational constraints, and in
order to obtain accurate distribution costs and guarantee feasible distribution de-
cisions, routing is considered in each time period of the coarse time grid. Note that
the MPRP is more involved than the classical PRP because of the added complex-
ity on the production side. For solving large instances of the MPRP, we propose an
MILP-based heuristic approach that relies on applying the integrated MILP model
and a dynamic route generation procedure in an iterative fashion. The effectiveness
of the proposed solution method is demonstrated in an extensive computational
study as well as in a real-world industrial test case.

7.2. Problem Statement

We consider a multicommodity supply chain that is characterized by a set of prod-
ucts $i \in I$, a set of production plants $p \in P$, of which each can produce all or a
subset of the products, and a number of product-specific customers, of which each
customer $c \in C_i$ has a given demand and storage capacity for product $i$.

We assume that each production plant can operate in a set of discrete operat-
ing modes $m \in M_p$, where each mode is defined by its production capacity and
cost function. The complexity in the production process arises from the fact that
generally, the products cannot be produced independently from each other; hence,
correlations in production rates have to be considered. Furthermore, the dynamic
behavior of the plant is constrained by restrictions on the rate of change and transi-
tions between operating modes. The plants have inventory capacities for storable
products.

Product-specific vehicles, e.g. tanker trucks, are used to transport products
from the plants to the customers. Each vehicle is assigned to one particular plant
and is defined by its capacity, speed, and cost, which may include fuel and labor
costs. For every trip, a vehicle leaves the plant, visits one or multiple customers,
and returns to the plant at the end of the trip. The length of a trip is limited.

The goal of the MPRP is to optimize production and distribution operations at
different levels of decision-making for a given planning horizon. On the produc-
tion side, the solution should provide a detailed production schedule involving the following decisions for every time period: the operating mode, the production rate for each product, and the amounts of products stored. On the distribution side, we want to make tactical decisions regarding plant-to-customer allocation; hence, we determine the amounts of products distributed from each plant to each customer and the assignment of vehicles to trips. Since feasibility has to be guaranteed, more detailed routing decisions may be obtained as a byproduct of the solution method; however, these decisions are not required since detailed routing will be subject to reoptimization in a separate subsequent step in which plant-to-customer allocation is fixed.

7.3. Model Formulation

We propose an MILP model for the MPRP, for which the mathematical formulation is presented in the following. Note that all continuous variables in this model are constrained to be nonnegative.

7.3.1. Multiscale Time Representation

In the proposed model, a discrete time representation is applied. While short-term operational decisions are made on the production side, mid-term tactical decisions are made on the distribution side; hence, two different time scales have to be considered. We create two time grids, one with a fine and the other with a coarse time discretization, where the time horizon is divided into time periods of the lengths $\Delta t^f$ and $\Delta t^c$, respectively, with $\Delta t^c$ chosen to be a multiple of $\Delta t^f$. For the sake of clarity, we refer to a time period in the fine time grid as a level-1 time period and to a time period in the coarse time grid as a level-2 time period whenever this distinction is necessary. Also, the notation is such that time period $t$ starts at time point $t-1$ and ends at time point $t$.

In the fine time grid, the planning horizon is defined by the set of time periods $T^f = \{1, 2, \ldots, \hat{t}^f\}$, a subset of $T^f = \{-\theta^{\max} + 1, -\theta^{\max} + 2, \ldots, 0, 1, \ldots, \hat{t}^f\}$, which also includes time periods in the past that are used in some constraints involving mode transition variables. The coarse time discretization creates the following two sets of time periods: $T^c = \{1, 2, \ldots, \hat{t}^c\}$ and $T^c = \{0, 1, \ldots, \hat{t}^c\}$. Furthermore, we define a set $\bar{T} = \{1, \Delta t^c/\Delta t^f + 1, 2\Delta t^c/\Delta t^f + 1, \ldots, (\hat{t}^c - 1)\Delta t^c/\Delta t^f + 1\}$, which is the set of level-1 time periods that begin at the same time points as the corresponding level-2 time periods.
We illustrate the notation with the example shown in Figure 7.1. Here, we have a planning horizon of 12 h, \( \Delta t^f = 1 \text{ h} \), \( \Delta t^c = 4 \text{ h} \); consequently, \( \bar{t}^f = 12 \), \( \bar{t}^c = 3 \), and \( \bar{T} = \{1, 5, 9\} \). Also, we have \( g^{\text{max}} = 4 \). Note that \( \bar{T}^f \) and \( \bar{T}^c \) only refer to the planning horizon starting with time period 1.

\[ \begin{align*}
\bar{T}^f &= \{1, 5, 9\} \\
\bar{T}^c &= \{0, 1, 2, 3\}
\end{align*} \]

**Figure 7.1:** Fine and coarse time grids for a planning horizon of 12 h with \( \Delta t^f = 1 \text{ h} \) and \( \Delta t^c = 4 \text{ h} \).

### 7.3.2. Production Scheduling

The proposed scheduling model is based on the model developed in Chapter 3; hence, we only provide brief descriptions of the familiar constraints.

**Scheduling Constraints**

The CRS models for the production plants in the supply chain are represented by the following set of constraints:

\[
P D_{pt} = \sum_{m \in M_p} \sum_{r \in R_{pm}} P D_{pmrit} \quad \forall \ p, \ i, \ t \in T^f \tag{7.1a}
\]

\[
P D_{pmrit} = \sum_{j \in J_{pmr}} \lambda_{pmrjt} v_{pmrji} \quad \forall \ p, \ m \in M_p, \ r \in R_{pm}, \ i, \ t \in T^f \tag{7.1b}
\]

\[
\sum_{j \in J_{pmr}} \lambda_{pmrjt} = \bar{y}_{pmrt} \quad \forall \ p, \ m \in M_p, \ r \in R_{pm}, \ t \in T^f \tag{7.1c}
\]

\[
y_{pmf} = \sum_{r \in R_{pm}} \bar{y}_{pmr} \quad \forall \ p, \ m \in M_p, \ t \in T^f \tag{7.1d}
\]

\[
\sum_{m \in M_p} y_{pmf} = 1 \quad \forall \ p, \ t \in T^f \tag{7.1e}
\]

\[
y_{pmf} \in \{0, 1\} \quad \forall \ p, \ m \in M_p, \ t \in T^f \tag{7.1f}
\]
where $\mathcal{M}_p$ is the set of operating modes in which plant $p$ can operate, $R_{pm}$ is the set of subregions in mode $m \in \mathcal{M}_p$ and $J_{pmr}$ is the set of vertices of subregion $r \in R_{pm}$. The binary variable $y_{pmr}t$ equals 1 if mode $m \in \mathcal{M}_p$ is selected in time period $t$, whereas the binary variable $\bar{y}_{pmr}t$ equals 1 if subregion $r \in R_{pm}$ is selected in time period $t$. The amount of product $i$ produced at plant $p$ in time period $t$ is denoted by $PD_{pit}$. Associated with $PD_{pit}$ is the disaggregated variable $PD_{pmr}t$ for subregion $r \in R_{pm}$, which is expressed as a convex combination of the corresponding vertices, $v_{pmrji}$. Notice that Eqs. (7.1) are written for all $t \in T^f$, which refers to the fine time discretization.

Constraints on the rate of change within an operating mode and on transitions between different modes are shown in the following:

\[
- \bar{\Delta}_{pmi}^{\max} \leq \sum_{r \in R_{pm}} (\bar{P}_{pmr}t - \bar{P}_{pmr}t-1) \leq \Delta_{pmi}^{\max} \quad \forall \ p, m \in \mathcal{M}_p, i, t \in T^f \tag{7.2a}
\]

\[
\sum_{m' \in TR_{pm}} z_{pm'm,m,t-1} - \sum_{m' \in TR_{pm}} z_{pmm',t-1} = y_{pmr}t - y_{pm,t-1} \quad \forall \ p, m \in \mathcal{M}_p, t \in T^f \tag{7.2b}
\]

\[
y_{pmr}t \geq \sum_{k=1}^{\theta_{pmn'}} y_{pmn',t-k} \quad \forall \ p, (m, m') \in TR_{p}, t \in T^f \tag{7.2c}
\]

\[
z_{pmn',t-\theta_{pmn'm''}} = z_{pmn',t} \quad \forall \ p, (m, m', m'') \in SQ_{p}, t \in T^f \tag{7.2d}
\]

\[
z_{pmn't} \in \{0, 1\} \quad \forall \ p, (m, m') \in TR_{p}, t \in T^f \tag{7.2e}
\]

where $\bar{\mathcal{T}}R_{pm} = \{m' : (m', m) \in TR_{p}\}$ and $\mathcal{T}R_{pm} = \{m' : (m, m') \in TR_{p}\}$ with $TR_{p}$ being the set of all possible mode-to-mode transitions at plant $p$, $SQ_{p}$ is the set of predefined sequences for plant $p$, $\theta_{pmn'}$ being the minimum stay time in mode $m' \in \mathcal{M}_p$ after switching to it from mode $m \in \mathcal{M}_p$, and $\bar{\theta}_{pmn'm''}$ is the fixed stay time in mode $m'$ in the corresponding sequence. The binary variable $z_{pmn't}$ equals 1 if and only if plant $p$ switches from mode $m \in \mathcal{M}_p$ to mode $m' \in \mathcal{M}_p$ at time $t$.

The following equations fix the initial mode of each plant according to the parameters $y_{pm}^{ini}$ and include the required information on the mode switching history in the form of the parameters $z_{pmn't}^{ini}$:

\[
y_{pm,0} = y_{pm}^{ini} \quad \forall \ p, m \in \mathcal{M}_p \tag{7.3a}
\]

\[
z_{pmn't} = z_{pmn't}^{ini} \quad \forall \ p, (m, m') \in TR_{p}, t \in T^f, -\bar{\Delta}_{pmi}^{\max} + 1 \leq t \leq -1 \tag{7.3b}
\]
with \( \bar{\theta}_p^{\text{max}} = \max \left( \max_{(m,m') \in TR_p} \{ \theta_{pmm'} \}, \max_{(m,m',m'') \in SQ_p} \{ \bar{\theta}_{pmm'm''} \} \right) \). Note that the fine time discretization can then be established by using \( \bar{\theta}_p = \max_p \{ \bar{\theta}_p^{\text{max}} \} \).

**Inventory Constraints**

First, we distinguish between storable and nonstorable products by creating the two disjoint product sets \( \bar{I} \) and \( \hat{I} \), respectively. While in general, storable products have to be transported to the customer locations, demands for nonstorable products are assumed to occur at the production plants. Therefore, for nonstorable products, it suffices to simply constrain the production to be higher than the demand:

\[
PD_{pit} \geq \bar{D}_{pit} \quad \forall \ p, i \in \hat{I}, t \in T^f
\]

(7.4)

where \( \bar{D}_{pit} \) denotes the demand for product \( i \) at plant \( p \) in time period \( t \).

Formulating the inventory constraints for the storable and therefore transportable products requires the following assumption: The products distributed to the customers in each level-2 time period are loaded into the vehicles within the first \( \Delta t_f \) of the same level-2 time period. This restriction is necessary due to the multiple time scales and it ensures that we always have sufficient inventory such that vehicles can leave the plants close to the beginning of the time period; otherwise, the vehicles may not be able to complete their trips within the same time period. With this assumption, we arrive at the following inventory constraints:

\[
IV_{pit} = IV_{pi,t-1} + PD_{pit} - LD_{ipt'} \quad \forall \ p, i \in \hat{I}, t \in T, t' = \pi_t
\]

(7.5a)

\[
IV_{pit} = IV_{pi,t-1} + PD_{pit} \quad \forall \ p, i \in \bar{I}, t \in T^f \setminus \bar{T}
\]

(7.5b)

\[
IV_{pit}^{\text{min}} \leq IV_{pit} \leq IV_{pit}^{\text{max}} \quad \forall \ p, i \in \bar{I}, t \in T^f
\]

(7.5c)

\[
IV_{pi,0} = IV_{pi}^{\text{ini}} \quad \forall \ p, i \in \bar{I}
\]

(7.5d)

where \( IV_{pit} \) is the inventory level of product \( i \) at plant \( p \) at level-1 time point \( t \), and \( LD_{ipt'} \) is the amount of product \( i \) loaded into vehicles at plant \( p \) in level-2 time period \( t' \). Since \( IV_{pit} \) and \( LD_{ipt'} \) refer to time periods in different time grids, they need to be matched, which is achieved by introducing the parameter \( \pi_t \), which denotes the level-2 time period that begins at the same time point as level-1 time period \( t \). Eq. (7.5a) states that the inventory level at time point \( t \) is the inventory level at the previous time point plus the amount of product produced in time period \( t \).
minus the amount loaded into vehicles in the same time period. Eq. (7.5b) tracks
the inventory in time periods in which no product is drawn from the storage to be
loaded into vehicles. Eq. (7.5c) sets lower and upper bounds on the inventory lev-
els, denoted by $IV_{	ext{min}}^{\text{pit}}$ and $IV_{	ext{max}}^{\text{pit}}$, respectively. Eq. (7.5d) fixes the initial inventory
level to the value of the parameter $IV_{\text{ini}}^{\text{pi}}$.

7.3.3. Distribution Planning

For the modeling of the distribution planning part of the problem, we make the
following assumptions:

- Vehicles that transport the same product have the same capacity, speed, and
  route-specific costs.
- At the end of every trip, a vehicle returns to its assigned production plant.
- Every trip is completed within a level-2 time period.
- A vehicle cannot make more than one trip in each level-2 time period.
- In each level-2 time period, a particular route can only be used by one vehicle.
- A customer can be visited by multiple vehicles in the same level-2 time pe-
  riod.

With these assumptions, we essentially have to solve in each level-2 time period a
distance-constrained capacitated VRP (DCVRP) where each customer can be vis-
itied by multiple vehicles from multiple plants. However, we also manage the cus-
tomers’ inventories. As a result, we do not have fixed orders; instead, the amounts
of products distributed to the customers are variables and therefore subject to op-
timization. Therefore, the level-2 time periods are coupled by the customer inven-
tories, leading to an IRP over the entire planning horizon. Note that all following
distribution planning constraints are formulated with respect to the coarse level-2
time discretization.

Flow Conservation Constraints

For each product $i$ and time period $t$, flow conservation has to be satisfied at every
node in the network representation of the distribution model, as depicted in Figure
7.2.
We apply a set-partitioning formulation (Balinski & Quandt, 1964) in which a set of feasible transportation routes is used where each route is defined as the set of customers that can be visited on the route. The resulting flow conservation constraints are as follows:

\[
LD_{ipt} = \sum_{s \in S_{ipt}} DL_{ipts} \quad \forall i \in \bar{I}, \ p, \ t \in T^c
\]  

(7.6a)

\[
DL_{ipts} = \sum_{c \in \bar{C}_{ips}} DL_{iptsc} \quad \forall i \in \bar{I}, \ p, \ t \in T^c, \ s \in S_{ipt}
\]  

(7.6b)

\[
DL_{ict} = \sum_{p} \sum_{s \in S_{ipt}} DL_{iptsc} \quad \forall i \in \bar{I}, \ c \in C_i, \ t \in T^c
\]  

(7.6c)

where for time period \( t \), \( DL_{ipts} \) denotes the amount of product \( i \) delivered on route \( s \) by a vehicle from plant \( p \), \( DL_{iptsc} \) is the amount delivered to customer \( c \) on route \( s \), and \( DL_{ict} \) is the total amount of product \( i \) delivered to customer \( c \). While \( S_{ip} \) denotes the set of routes that can be used by vehicles assigned to plant \( p \), \( S_{ipt} \) is the subset of \( S_{ip} \) that can be used in time period \( t \). The set of customers that can be visited on route \( s \in S_{ip} \) is denoted by \( \bar{C}_{ips} \).
7. Multiscale Production Routing in Power-intensive Supply Chains

Capacity Constraints

The distribution resource constraints in terms of the vehicle capacity and the available number of vehicles are stated as follows:

\[
\overline{DL}_{ipts} \leq V_i \cdot x_{ipts} \quad \forall \; i \in \tilde{I}, \; p, \; t \in \tilde{T}, \; s \in S_{ipt} \quad (7.7a)
\]
\[
\overline{DL}_{iptsc} \leq \overline{DL}_{i\text{ct}}^{\text{max}} \cdot x_{ipts} \quad \forall \; i \in \tilde{I}, \; p, \; t \in \tilde{T}, \; s \in S_{ipt}, \; c \in \overline{C}_{ips} \quad (7.7b)
\]
\[
\sum_{s \in \overline{S}_{ipt}} x_{ipts} \leq L_{ipt} \quad \forall \; i \in \tilde{I}, \; p, \; t \in \tilde{T} \quad (7.7c)
\]
\[
x_{ipts} \in \{0, 1\} \quad \forall \; i \in \tilde{I}, \; p, \; t \in \tilde{T}, \; s \in \overline{S}_{ipt} \quad (7.7d)
\]

where \(V_i\) is the capacity of a vehicle transporting product \(i\), and \(\overline{DL}_{i\text{ct}}^{\text{max}}\) can be set to \(\min \{V_i, \overline{V}_{i\text{ct}}^{\text{max}} - \overline{V}_{i\text{ct},t-1}^{\text{low}} + \overline{D}_{i\text{ct}}, \sum_{t' \in \tilde{T}} \overline{D}_{i\text{ct}}, + \overline{V}_{i\text{ct},t-1}^{\text{low}} - \overline{V}_{i\text{ct},t-1}^{\text{low}}\}\) with \(\overline{V}_{i\text{ct},t-1}^{\text{low}} = \max \{\overline{V}_{i\text{ct},t-1}^{\text{min}}, \overline{V}_{i\text{ct},0} - \sum_{t' \in \tilde{T}} \overline{D}_{i\text{ct}}\}\), which is the lowest possible inventory level at time \(t - 1\). The binary variable \(x_{ipts}\) equals 1 if a vehicle from plant \(p\) transporting product \(i\) takes route \(s\) in time period \(t\). Eqs. (7.7a)–(7.7b) set upper bounds on the distribution variables and force them to zero if the corresponding routes are not selected. In Eq. (7.7c), \(L_{ipt}\) denotes the number of vehicles that transport product \(i\) and are available at plant \(p\) in time period \(t\).

Inventory Constraints

The constraints on the inventories at the customer sites are formulated as follows:

\[
\overline{IV}_{i\text{ct}} = \overline{IV}_{i\text{ct},t-1} + DL_{i\text{ct}} + PC_{i\text{ct}} - D_{i\text{ct}} \quad \forall \; i \in \tilde{I}, \; c \in C_i, \; t \in \tilde{T} \quad (7.8a)
\]
\[
\overline{IV}_{i\text{ct}}^{\text{min}} \leq \overline{IV}_{i\text{ct}} \leq \overline{IV}_{i\text{ct}}^{\text{max}} \quad \forall \; i \in \tilde{I}, \; c \in C_i, \; t \in \tilde{T} \quad (7.8b)
\]
\[
\overline{IV}_{i\text{ct},0} = \overline{IV}_{i\text{ct}}^{\text{ini}} \quad \forall \; i \in \tilde{I}, \; c \in C_i \quad (7.8c)
\]

where \(\overline{IV}_{i\text{ct}}\) denotes the inventory level for product \(i\) at customer \(c\) at time point \(t\), and \(PC_{i\text{ct}}\) is the amount of product purchased externally in case the demand, denoted by \(D_{i\text{ct}}\), cannot be satisfied by drawing from the own inventory. Lower and upper bounds on \(\overline{IV}_{i\text{ct}}\) are denoted by \(\overline{IV}_{i\text{ct}}^{\text{min}}\) and \(\overline{IV}_{i\text{ct}}^{\text{max}}\), respectively, and \(\overline{IV}_{i\text{ct}}^{\text{ini}}\) is the initial inventory.

7.3.4. Objective Function

The objective is to minimize the total operating cost, \(TC\), consisting of production costs, purchasing costs, distribution costs, and inventory costs; hence, the objective
function is:

\[
TC = \sum_{p \in M_p} \sum_{r \in R_{pm}} \sum_{t \in T_p^r} \left( \delta_{pmrt} \bar{y}_{pmrt} + \sum_i \gamma_{pmrit} PD_{pmrit} \right) \\
+ \sum_{i \in I} \sum_{t \in T_{c_i}} \sum_{c \in C_i} \alpha_{ict} PC_{ict} + \sum_{i \in I} \sum_{t \in T_{c_i}} \sum_{s \in S_{ipt}} \beta_{ips} x_{ipts} \\
+ \sum_{p \in P} \sum_{i \in I} \sum_{t \in T_p} \rho_{pit} IV_{pit} + \sum_{i \in I} \sum_{c \in C_i} \sum_{t \in T_{c_i}} \bar{\rho}_{ict} IV_{ict}
\]

(7.9)

where \(\delta_{pmrt}\) and \(\gamma_{pmrit}\) are the fixed and unit production costs, respectively, in operating subregion \(r \in R_{pm}\) in level-1 time period \(t\). The unit cost for purchasing product \(i\) to satisfy demand at customer \(c\) in level-2 time period \(t\) is \(\alpha_{ict}\). The fixed distribution cost for using route \(s \in S_{ip}\) is \(\beta_{ips}\). The unit inventory costs for storing product \(i\) in time period \(t\) at plant \(p\) and customer \(c\) are denoted by \(\rho_{ict}\) and \(\bar{\rho}_{ict}\), respectively. With this objective function, the MILP for the MPRP then becomes:

\[
\text{min } TC \\
\text{s.t. } \text{Eqs. (7.1)} - (7.9).
\]

(7.4) Solution Method

The difficulty in solving (MPRP) is mainly due to the integration of two very complex problems: a detailed MILP production scheduling problem and an IRP with high combinatorial complexity. Especially in such multiplant multicommodity supply chains, the interdependencies are very strong and have to be taken into account in order to obtain good solutions. In the following, we propose an MILP-based heuristic solution method involving dynamic route generation, which is designed to solve MPRPs of industrially relevant sizes.

In the distribution part of the proposed MPRP model, a set-partitioning formulation is applied where routing decisions are made by selecting a set of feasible routes. Note that a route is considered feasible if the trip time does not exceed \(\bar{\tau}_{max}\), which is typically set to \(\Delta t_c\). This kind of formulation is known to exhibit a relatively tight LP relaxation, but it can require an exponential number of routes to fully describe the problem. However, at a feasible solution, only a very small fraction of all possible routes are selected. Hence, instead of working with the full route set, we propose to only consider a small subset of routes when solving (MPRP) and dynamically update the route set such that only good candidate routes
are included. An outline of the proposed algorithm is as follows:

**Step 1** For each product $i$ and plant $p$, create an initial set of routes, $S_{ip}$. Each route $s \in S_{ip}$ is defined by the set of customers that can be reached on this route, $C_{ ips }$, and the fixed distribution cost, $\beta_{ips}$. Furthermore, for each level-2 time period $t$, create $\mathcal{S}_{ipt}$, which is the subset of $S_{ip}$ that is considered in time period $t$.

**Step 2** Solve (MPRP) with the current set of candidate routes.

**Step 3** Based on the solution of (MPRP), add new routes to or remove existing routes from the current route set, i.e. update all $S_{ip}$, $C_{ips}$, $\beta_{ips}$, and $\mathcal{S}_{ipt}$.

**Step 4** If a stopping criterion is satisfied, stop; otherwise, go to Step 2.

Since only a subset of all possible routes is considered when solving (MPRP) in Step 2, the computational complexity is reduced, but we are likely to only obtain a suboptimal solution. Inefficiencies on the distribution side are treated in Step 3 by updating the route set such that it includes candidate routes that can potentially improve the solution. The selection of new candidate routes is based on a local analysis of the current solution, i.e. it does not consider all relationships that exist in the integrated problem. Therefore, instead of directly applying a new route to improve the current solution, we decide whether the proposed route should be selected by solving (MPRP) in the next iteration.

The proposed solution algorithm is inspired by the concept of column generation, with the main difference being that here, new columns are generated by using a heuristic rather than by solving a rigorous pricing problem. In the following subsections, the major steps of the algorithm are described in more detail. As we will show, the solution is guaranteed to improve or at least remain the same at each iteration if (MPRP) is always solved to optimality. However, convergence to the optimal solution cannot be guaranteed, which is the main limitation of the proposed algorithm.

### 7.4.1. Initialization

For the initial set of routes, we may consider all single-stop routes, i.e. only one customer can be visited in each trip; however, in large-scale instances, even this route set can be prohibitively large. We realize that in most practical applications, the vast majority of the customers are only visited in a few time periods over the
planning horizon. Hence, in order to reduce the number of single-stop routes considered in the initial iteration, we determine for each customer \( c \in C_i \) the time periods in which it will likely be receiving delivery, and denote this set of time periods by \( T_{ic}^{del} \). We then only consider feasible single-stop routes to customer \( c \) in these time periods as well as in the \( \omega \) previous and \( \omega \) following time periods, i.e. in time periods \( t \) such that \( t' - \omega \leq t \leq t' + \omega \) where \( t' \in T_{ic}^{del} \). By changing the parameter \( \omega \), we can adjust the number of routes included in the initial route set. We propose to determine \( T_{ic}^{del} \) as follows: Apply an inventory policy in which a customer’s inventory is refilled to its maximum level, \( IV_{ict}^{max} \), in time period \( t \) if otherwise the inventory level falls below \( IV_{ict}^{min} \) at the end of time period \( t \). Choose \( T_{ic}^{del} \) to be the set of replenishment points.

### 7.4.2. Updating Set of Candidate Routes

Algorithm 1 shows the general scheme for generating routes based on the current solution of (MPRP), which may not be optimal or near-optimal (especially in the first iteration), but provides a good estimate of the amount of product that needs to be delivered to each customer in each time period. Using this information, the algorithm identifies inefficiencies in the current selection of routes and proposes new candidate routes that may improve the solution.

```latex
\begin{algorithm}
\begin{enumerate}
\item for all \( i \in I, p, t \in T^c \) do
\item \hspace{1em} REMOVE ROUTES \((i, p, t, \Omega)\)
\item \hspace{1em} for all \( s \) for which \( x_{ipts} = 1 \) and \( DL_{ipts} < V_i \) do
\item \hspace{2em} CREATE ROUTES A \((i, p, t, s)\)
\item \hspace{1em} end for
\item \hspace{1em} for all \( c \in C_{ip} \) for which \( PC_{ict} > 0 \) do
\item \hspace{2em} CREATE ROUTES B \((i, p, t, c)\)
\item \hspace{1em} end for
\item end for
\end{enumerate}
\end{algorithm}
```

At each iteration, the algorithm is applied to every product \( i \in I \), plant \( p \), and time period \( t \in T^c \). First, the procedure REMOVE ROUTES \((i, p, t, \Omega)\) removes routes that have not been selected for \( \Omega \) consecutive iterations from the set \( S_{ipt} \). Next, the distribution inefficiency due to underutilized vehicles is considered. We examine every selected route \( s \) for which the delivery quantity is less than the vehicle capacity, i.e. \( DL_{ipts} < V_i \). The procedure CREATE ROUTES A \((i, p, t, s)\) generates new routes, if possible, by inserting additional customers into the current route \( s \). A se-
lection of these new routes are added to the route set $S_{ipt}$ based on a ranking of the potential savings. Besides underutilized vehicles, another indicator for distribution inefficiency is the purchase of products at high costs, which usually occurs due to the lack of efficient multistop routes. Hence, in the next step, we consider customers whose demands are met by purchasing additional products, i.e. all $c \in \tilde{C}_{ip}$ for which $PC_{ict} > 0$, where $\tilde{C}_{ip}$ is a subset of $C_i$ and denotes the the set of customers that can be reached from plant $p$. Similar to $\text{CREATE ROUTES}A(i, p, t, s)$, the procedure $\text{CREATE ROUTES}B(i, p, t, c)$ generates multistop routes involving customer $c$ and adds them to $S_{ipt}$ based on a ranking of the potential savings.

In Algorithm 2, we describe the procedure $\text{CREATE ROUTES}A(i, p, t, s)$ in more detail. As stated in lines 1–2, we first choose $C_{\text{del}}$, which is the set of customers to which delivery on a new route is considered. A customer is included in $C_{\text{del}}$ if it can be reached from plant $p$, is not already part of route $s$, and is expected to receive delivery in time period $t$ or any of the $\xi$ subsequent time periods. We consider the latter condition to be satisfied if $PC_{ict'} > 0$ or $DL_{ict'} > 0$ for any $t'$ between $t$ and $t + \xi$. The parameter $\xi$ can be adjusted to control the number of customers considered. The size of $C_{\text{del}}$ increases with increasing $\xi$, and hence the computational effort increases; however, it has the benefit of making the search for better routes less localized. In lines 4–6, $S_{\text{pot}}$ and $S_{\text{check}}$ are initialized with the current route $s$. While $S_{\text{pot}}$ is the set of potential new routes, $S_{\text{check}}$ is the subset of $S_{\text{pot}}$ that need to be further examined because more customers may be included in these routes. In general, the procedure $\text{ADDTOUR}(\tilde{C}, \tilde{\beta}, \tilde{S}, \tilde{n})$ adds the route characterized by customer set $\tilde{C}$ and distribution cost $\tilde{\beta}$ to the route set $\tilde{S}$, where the new route is indexed by $\tilde{n}$.

For each $s' \in S_{\text{check}}$, $c \in C_{\text{del}}$, we check whether by inserting customer $c$ into route $s'$ results in a new feasible route. By executing the procedure $\text{COMPUTE TSP}(i, p, \tilde{C})$, the traveling salesman problem (TSP) is solved, which provides the minimum travel time, $\tau_{\text{travel}}$, for a vehicle to transport product $i$ from plant $p$ to all customers in $\tilde{C}$ and returning to the same plant at the end of the trip. In addition to the travel time, the time spent on a trip also includes the time that the vehicle stays at each location for the purpose of loading and unloading; hence, the total trip time is $\bar{\tau} = \tau_{\text{travel}} + \tau_{\text{stay}}(|\tilde{C}| + 1)$, where $\tau_{\text{stay}}$ is the average time that a vehicle transporting product $i$ spends at each location. A route is feasible if $\bar{\tau} \leq \bar{\tau}_{\text{max}}$.

If the potential new route is feasible, the distribution cost is computed, and the route is added to the route set $S_{\text{pot}}$ (see lines 11–13). Here, $\beta_{\text{travel}}$ and $\beta_{\text{stay}}$ denote the unit travel cost and the fixed cost for loading and unloading, respectively. Then
Algorithm 2 Create new routes based on route selected in the current solution.

1: procedure CREATE RoutesA(i, p, t, s)
2: \[ C_{\text{del}} \leftarrow \{ c : \sum_{t' = t}^{t + \xi} PC_{ict'} > 0, c \in \tilde{C}_{ip}, c \notin \tilde{C}_{ips} \} \]
3: \[ C_{\text{del}} \leftarrow C_{\text{del}} \cup \{ c : \sum_{t' = t}^{t + \xi} DL_{ict'} > 0, c \in \tilde{C}_{ip}, c \notin \tilde{C}_{ips} \} \]
4: \[ \text{Spot} \leftarrow \emptyset, n_{\text{pot}} \leftarrow 1 \]
5: ADDROUTE(\text{Spot, } \beta_{\text{ips}}, S_{\text{pot}}, n_{\text{bot}})
6: \[ S_{\text{check}} \leftarrow \{ n_{\text{pot}} \} \]
7: for all \( s' \in S_{\text{check}} \) do
8: \[ \tilde{C} \leftarrow C_{s'} \cup \{ c \} \]
9: \[ \tau_{\text{travel}} \leftarrow \text{COMPUTE TSP}(i, p, \tilde{C}) \]
10: if \( \tau_{\text{travel}} + \tau_{\text{stay}}(|\tilde{C}| + 1) \leq \tau_{\text{max}} \) then
11: \[ n_{\text{pot}} \leftarrow n_{\text{pot}} + 1 \]
12: \[ \beta \leftarrow \beta_{\text{ips}} \tau_{\text{travel}} + \beta_{\text{ips}}^{' t}(|\tilde{C}_{ips}| + 2) \]
13: ADDROUTE(\text{C}, \beta, S_{\text{pot}}, n_{\text{bot}})
14: \[ DL_{n_{\text{pot}}}^{\text{pot}} \leftarrow \text{COMPUTE LOAD}(n_{\text{pot}}, s') \]
15: \[ SAV_{\text{pot}}^{\text{pot}} \leftarrow \text{COMPUTE SAVINGS}(n_{\text{pot}}, s') \]
16: if \( DL_{n_{\text{pot}}}^{\text{pot}} < V_i \) and \( |C_{n_{\text{pot}}}^{\text{pot}}| < N_{\text{cmax}} \) then
17: \[ S_{\text{check}} \leftarrow S_{\text{check}} \cup \{ n_{\text{pot}} \} \]
18: end if
19: end if
20: end for
21: RANK AND ADD(S_{\text{pot}}, S_{\text{ipt}}, N_{\text{smax}})
22: end procedure

we apply the procedure COMPUTE LOAD(n_{pot}, s') to compute \( DL_{n_{pot}}^{\text{pot}} \), which is an estimate of the vehicle load if route \( n_{pot} \) is used. This estimate is obtained by filling the vehicle used on route \( s' \) in a greedy fashion; for instance, if \( \sum_{t' = t}^{t + \xi} PC_{ict'} > 0 \), then \( DL_{n_{pot}}^{\text{pot}} = \min \{ V_i, DL_{s'}^{\text{pot}} + \sum_{t' = t}^{t + \xi} PC_{ict'} \} \). Under this vehicle load assumption, the savings of taking route \( n_{pot} \) instead of \( s' \), \( SAV_{n_{pot}}^{\text{pot}} \), can be computed by using the procedure COMPUTE SAVINGS(n_{pot}, s'). If there is still remaining capacity in the vehicle, i.e. \( DL_{n_{pot}}^{\text{pot}} < V_i \), and the number of customers on route \( n_{pot} \) has not reached the set maximum, \( N_{\text{cmax}} \), \( n_{pot} \) is added to \( S_{\text{check}} \) such that it can be further examined and extended to another new route if possible. Finally, after the set of potential new routes, \( S_{\text{pot}} \), is generated, CREATE RoutesA(i, p, t, s) is completed by the procedure RANK AND ADD(S_{pot}, S_{ipt}, N_{smax}), which ranks all routes in \( S_{\text{pot}} \) according to their potential savings and adds the top \( N_{\text{smax}} \) routes to \( S_{ipt} \).

The route generation algorithm in CREATE RoutesB(i, p, t, c) is very similar to the one in CREATE RoutesA(i, p, t, s). The main difference is that CREATE RoutesB(i, p, t, c) considers the single-top route from \( p \) to \( c \) as the initial route in
$S_{pot}^*$, while CREATE ROUTES A($i,p,t,s$) initializes $S_{pot}^*$ with route $s$.

### 7.4.3. Stopping Criteria

Let $TC^k$ be the total cost value obtained by solving (MPRP) in iteration $k$. If (MPRP) is solved to optimality in every iteration, then $TC^k \geq TC^{k+1}$, i.e. the objective function value is guaranteed to improve or remain the same at each iteration. This statement holds since the routes selected in iteration $k$ remain in the route set considered in iteration $k + 1$, i.e. the optimal solution of (MPRP) in iteration $k$ is a feasible solution of (MPRP) in iteration $k + 1$.

In the form as it is presented here, the proposed algorithm does not guarantee convergence to the optimal solution. It can be modified such that at some point, all possible routes are included in the model. In that case, the algorithm would converge to the optimal solution; however, such an implementation has little practical value since industrial-scale instances of (MPRP) with all possible routes cannot be solved in a reasonable time. Our goal is to obtain good solutions in short computation times; hence, besides setting a time limit, we propose to terminate the algorithm when one of the following two stopping criteria applies:

1. The relative improvement in the objective function from one iteration to the next, defined as $(TC^k - TC^{k+1})/TC^k$, has been less than $\epsilon$ for $\Phi$ consecutive iterations.

2. Less than $\Psi$ new routes have been generated in the current iteration.

### 7.4.4. Algorithmic Parameters

In our computational experiments, the proposed algorithm has proven to be very robust with regard to the algorithmic parameters. In the following, we list all required parameters and provide guidelines for their settings:

- $\omega$ - number of time periods preceding and following time periods in $T_{del}$ that are considered in the alternative initial single-stop route generation procedure; we recommend setting $\omega$ to an integer between 0 and 3.

- $\Omega$ - number of consecutive iterations in which a route has not been used before it is deleted from the route set, typically set to 1 for the first iteration and 2 for all remaining iterations.
• $N_{c_{\text{max}}}$ - maximum number of customers considered on a new route; we recommend setting $N_{c_{\text{max}}}$ such that it increases with each iteration until it reaches the maximum number of customers at which efficient routes can still be expected, such a gradual increase in $N_{c_{\text{max}}}$ prevents the algorithm from getting trapped in a local solution too quickly.

• $N_{s_{\text{max}}}$ - maximum number of new routes added to the route set after one run of CREATE ROUTES A($i, p, t, s$) or CREATE ROUTES B($i, p, t, c$), should be as large as the computational budget allows.

• $\xi$ - number of subsequent time periods from which customers with deliveries or product purchases can be considered on a new route for the current time period, typically set to 1 or 2.

• $\epsilon$ - relative change between the costs from two consecutive iterations below which the improvement is considered insignificant; we recommend setting $\epsilon$ to a number between 0.001 and 0.01.

• $\Phi$ - number of consecutive iterations with no significant improvement after which the algorithm terminates, typically set to 2.

• $\Psi$ - number of new routes below which the algorithm terminates, can be conservatively set to 5 or 10 in most applications.

7.5. Numerical Results

In the following, we use an illustrative example to demonstrate the main features of the proposed framework, test the algorithm’s performance in an extensive computational study, and apply it to a real-world industrial gas supply chain. All models were implemented in GAMS 24.4.6 (GAMS Development Corporation, 2015b), and the commercial solver CPLEX 12.6.2 (IBM ILOG, 2015b) was applied to solve the MILPs on an Intel® Core™ i7-4770 machine at 3.40 GHz with 8 processors and 16 GB RAM running Windows 7 Enterprise.

In all instances, we set $\Omega = \min\{k, 2\}$ with $k$ being the iteration counter, $\epsilon = 0.001$, $\Phi = 2$, $\Psi = 5$ when applying the proposed algorithm. The choice of the other algorithmic parameters varies slightly across the different instances.
7.5.1. Illustrative Example

In the illustrative example, we consider a supply chain with two products, I1 and I2, two production plants, P1 and P2, and 50 customers, among which 20 require Product I1 and 30 require Product I2. The corresponding supply chain network is shown in Figure 7.3. Plant P1 has a fleet with 3 vehicles for Product I1 and 3 vehicles for Product I2; Plant P2 has 3 vehicles for Product I1 and 4 vehicles for Product I2.

Figure 7.3: Supply chain network for the illustrative example, with 2 plants and 50 customers.

The feasible regions of the given production modes are shown in Figure 7.4. Plant P1 can only operate in one mode, P1-M1, whereas Plant P2 can operate in three different modes, P2-M1, P2-M2, and P2-M3. Plant P2 cannot directly switch from Mode P2-M1 to Mode P2-M2; instead, it has to transition through the intermediate mode P2-M3. Note that Mode P2-M3 is described by a single operating point.

A scheduling horizon of 36 h is considered. We set $\Delta t_f = 1$ and $\Delta t_c = 12$, resulting in 36 level-1 and 3 level-2 time periods. The resulting MPRP has 9192 continuous variables, 2486 binary variables, and 11,197 constraints, and is solved to optimality in about 10 min. Figures 7.5 and 7.6 show the product flows and inventory profiles for both products at Plants P1 and P2, respectively. Note that in the figures, the y-axes for the inventory levels and the product flows are shown on...
the left and right hand sides, respectively. Positive columns (production) indicate accumulation of products in the inventory, while negative columns (shipments) indicate depletion of products. In overall, Plant P1 produces more than Plant P2 because of its lower unit production cost.

The optimal routing decisions are shown in Figure 7.7, where each subfigure refers to one of the 3 level-2 time periods and shows the selected routes including the corresponding delivery quantities. The load capacity of each vehicle is 500 kg. As one can see, the solution suggests loading the vehicles as close to full capacity as possible. Also, not all vehicles are used in every time period. The total number of selected routes is 22.

We now apply the proposed heuristic solution algorithm to this illustrative example. The algorithmic parameters are set as follows: $\xi = 1$, $N_{c_{\text{max}}} = \min\{k + 1, 4\}$ where $k$ is the iteration counter, and $N_{s_{\text{max}}} = 4$. Furthermore, all possible single-stop routes are considered in the first iteration.

To illustrate the evolution of the solution from one iteration to the next, we show the routing decisions from the first three iterations in Figure 7.8. At Iteration 1 (Figure 7.8a), only single-stop routes are considered, resulting in the dispatch of a large number of vehicles, most of which only carry a fraction of the maximum possible load. A total number of 35 routes are selected in this initial solution. At
Figure 7.5: Production quantities, shipments, and inventory levels of products I1 and I2 at Plant P1 in the illustrative example.

Iteration 2 (Figure 7.8b), new candidate routes are considered, which also include routes with two customers. The solution obtained at Iteration 2 is significantly more efficient in terms of distribution, as indicated by larger amounts of products delivered with fewer vehicles. The change in the routing decisions is smaller from Iteration 2 to Iteration 3 than from Iteration 1 to Iteration 2. However, one can see that the distribution plan has been further improved by considering also routes with three customers. In the solution obtained at Iteration 3, a total number of 23 routes are selected; recall that 22 routes are selected in the optimal solution.

In this case, the algorithm terminates after five iterations; however, the same solution is obtained at Iterations 3 to 5, although different candidate routes are considered. Note that no routes with four customers are selected although such routes are considered at Iterations 4 and 5. Table 7.1 compares the optimal solution with the solutions obtained at each iteration of the heuristic algorithm. The table shows the breakdown of the total costs ($TC$) into production costs ($CPD$), purchasing costs ($CPC$), distribution costs ($CDI$), inventory costs at the plants ($CIP$), and inventory costs at the customer sites ($CIC$). From the results of the heuristic
algorithm, one can clearly see that \( CPD \) increases while \( CPC \) and \( CDI \) decrease from one iteration to the next, which indicates that improved routing decisions are made at each iteration such that more products can be delivered from the plants and less has to be purchased from external sources. The algorithm terminates after no improvement is seen at Iterations 4 and 5. The final heuristic solution exhibits a total cost of $57,345, which is 1% higher than the total cost at the optimal solution.
$7. \text{ Multiscale Production Routing in Power-intensive Supply Chains}$

(a) Solution from Iteration 1, considering only single-stop routes.

(b) Solution from Iteration 2, considering routes with up to two customers.

(c) Solution from Iteration 3, considering routes with up to three customers.

Figure 7.8: Evolution of the routing solution obtained from applying the proposed heuristic algorithm to the illustrative example.

Table 7.1 further shows the number of candidate routes, $NR$, considered in each problem. This number is obtained by counting the routes for all products, plants, and time periods, i.e. $NR = \sum_{i \in I} \sum_{p} \sum_{t \in T} |S_{ipt}|$. One can see that in each iteration of the heuristic algorithm, a route set of significantly smaller size is considered compared with the full MPRP formulation. Because of this reduction in problem size through the dynamic route generation procedure, the near-optimal heuristic
Table 7.1: Comparison of costs and number of candidate routes in the full MPRP formulation and the restricted MPRPs solved in the heuristic algorithm.

<table>
<thead>
<tr>
<th></th>
<th>Optimal</th>
<th>Heuristic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iteration 1</td>
<td>Iteration 2</td>
</tr>
<tr>
<td>$TC\ [$]$</td>
<td>56,760</td>
<td>67,475</td>
</tr>
<tr>
<td>$CPD\ [$]$</td>
<td>27,975</td>
<td>25,282</td>
</tr>
<tr>
<td>$CPC\ [$]$</td>
<td>1703</td>
<td>13,140</td>
</tr>
<tr>
<td>$CDI\ [$]$</td>
<td>24,050</td>
<td>25,928</td>
</tr>
<tr>
<td>$CIP\ [$]$</td>
<td>615</td>
<td>518</td>
</tr>
<tr>
<td>$CIC\ [$]$</td>
<td>2418</td>
<td>2606</td>
</tr>
<tr>
<td>$NR$</td>
<td>2076</td>
<td>198</td>
</tr>
</tbody>
</table>

A solution was found in less than 20 s, which is a significant reduction in computation time compared to the 10 min required to solve the full MPRP model.

7.5.2. Computational Study

In the following, we test the computational performance of the proposed algorithm on a set of MPRP instances of different sizes.

Data Generation

For the computational study, we generate five sets of MPRP instances, Sets A to E, each containing ten instances of the same size. Table 7.2 lists for the instances in each data set the number of products, $|\tilde{I}|$ (only storable products are considered), number of plants, $|P|$, number of customers for each product $i$, $|C_i|$, number of vehicles across all plants for each product $i$, $\sum_p L_{ip}$ (with $L_{ip}$ being the number of vehicles that can transport product $i$ from plant $p$), number of level-1 time periods, $|\hat{T}|$, and number of level-2 time periods, $|\tilde{T}|$. Note that the ratio between $|\hat{T}|$ and $|\tilde{T}|$ is 12 in all instances.

The instances in each set differ in the customer locations, which are randomly generated on a 600 × 500 Euclidean grid, inventory capacities, initial inventory levels, and demands. While the customer demands are constant in the first five instances of each set, demands in the latter five instances vary over time.

Solution Methods

In the computational study, we compare the following four solution methods:
Table 7.2: Overview of generated MPRP instances, grouped into five data sets.

| Set | \( |\bar{I}| \) | \( |P| \) | \( |C_i| \) | \( \sum_p \bar{L}_{ip} \) | \( |T| \) | \( |\bar{T}| \) |
|-----|-------------|-------------|-------------|----------------|--------|--------|
| A   | 2           | 2           | 20          | 30             | 4      | 5      | -      | 36   | 3     |
| B   | 2           | 2           | 20          | 30             | 4      | 5      | -      | 120  | 10    |
| C   | 3           | 2           | 50          | 60             | 40     | 9      | 13     | 7    | 120   | 10    |
| D   | 3           | 2           | 50          | 60             | 40     | 9      | 13     | 7    | 360   | 30    |
| E   | 3           | 3           | 100         | 100            | 100    | 14     | 18     | 12   | 360   | 30    |

**Exact method** Solve (MPRP) considering all possible routes. Due to the computational limitations, the exact method is only applied to the instances in Sets A and B.

**Heuristic H1** This is a typical two-phase heuristic. In Phase 1, we solve the MPRP with a simplified distribution model only considering direct shipments. Here, we only consider all feasible single-stop routes, relax the integrality constraints on the variables \( x_{ipts} \), and solve (MPRP) where we replace Eq. (7.7c) by

\[
\sum_{s \in S_{ipt}} x_{ipts} \leq \eta L_{ipt} \quad \forall \ i \in \bar{I}, \ p, \ t \in \bar{T}
\]  

and add

\[
\sum_{s \in S_{ipt}} \tau_{ipts} x_{ipts} \leq \bar{\eta} L_{ipt} \bar{\tau}_{\max} \quad \forall \ i \in \bar{I}, \ p, \ t \in \bar{T}.
\]

Eqs. (7.10) and (7.11) are resource constraints on the total delivery quantity and travel time, respectively. Parameter \( \eta \) is typically set to a value between 0.8 and 1.2, while \( \bar{\eta} \) can be set to a value between 1 and 1.5. Setting \( \bar{\eta} > 0 \) accounts for the overestimation of the travel time in the model caused by only considering single-stop routes. For the instances considered in this computational study, setting both \( \eta \) and \( \bar{\eta} \) to 1 has proven to be a good choice.

The delivery quantities obtained from solving the simplified MPRP in Phase 1 are used as fixed orders in Phase 2, where routing decisions are made. Since the orders are fixed, the routing problem decomposes into independent subproblems, one for each product \( i \), plant \( p \), and time period \( t \in \bar{T} \). Each subproblem is a DCVVRP with the additional option of purchasing products if the...
orders cannot be met by delivering from the plant. We solve the DCVRPs with the MILP formulation shown in Appendix E. After solving the routing problem, the total cost is updated by replacing the distribution cost from Phase 1 with the routing cost obtained in Phase 2, adding the purchasing cost from Phase 2, and discounting the production cost associated with products that should be delivered according to Phase 1 but could not in Phase 2.

Although the vast majority of the DCVRPs considered in Phase 2 are very small in size (only a few customers) and can be solved within seconds, we set a time limit of 60 s to avoid stalling of the algorithm.

**Heuristic H2** Create a set of routes a priori and solve (MPRP) considering these candidate routes. The effectiveness of this solution strategy strongly depends on the number and quality of the generated routes. Here, we use the heuristic route generation procedure proposed by Marchetti et al. (2014), who have successfully applied this approach to industrial gas supply chain cases.

Marchetti et al. (2014) introduce four parameters: $c_{\text{max}}$, $s_{\text{max}}$, $v_{\text{min}}$, and $v_{\text{max}}$. The route generation procedure first generates all feasible routes with up to $c_{\text{max}}$ customers, and computes for each route a so-called logistics ratio, which is the ratio between the distribution cost and the maximum quantity that can be delivered on this route. The routes with the lowest logistics ratios are selected to be considered in (MPRP) such that, if possible, each customer can be visited on at least $v_{\text{min}}$ and not more than $v_{\text{max}}$ routes, and the number of routes for each product and plant is not larger than $s_{\text{max}}$. The resulting route set for each product and plant, $S_{ip}$, is considered in every time period, i.e. $S_{ipt} = S_{ip} \forall t \in T^c$.

Table 7.3 shows the parameter settings chosen in this computational study. For Sets A–D, two runs of Heuristic H2, denoted by H2a and H2b, are performed, where H2b considers more routes.

**Heuristic H3** Apply the MILP-based heuristic algorithm with dynamic route generation proposed in Section 7.4. The parameter settings for the different sets of instances are shown in Table 7.4. For solving the instances in Sets A–D, the algorithm is initialized with all possible single-stop routes, while for Set E, we create a smaller number of single-stop routes using the alternative procedure described in Section 7.4.1. For Sets B–E, we solve (MPRP) to 0.5 % optimality gap if possible, and further specify a time limit of 600 s for each MILP.
It should be mentioned that Heuristics H1 and H2 are solution approaches that are commonly used in practice for solving large-scale integrated supply chain problems like the MPRP; hence, we choose to compare the proposed algorithm, Heuristic H3, with these two solution strategies.

Table 7.3: Parameter settings for Heuristic H2.

<table>
<thead>
<tr>
<th>Set</th>
<th>Heuristic H2a/H2</th>
<th>Heuristic H2b</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cmax smax vmin vmax</td>
<td>cmax smax vmin vmax</td>
</tr>
<tr>
<td>A–D</td>
<td>4 200 2 5</td>
<td>4 200 5 10</td>
</tr>
<tr>
<td>E</td>
<td>3 200 2 5</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.4: Parameter settings for Heuristic H3, with $k$ being the iteration counter.

<table>
<thead>
<tr>
<th>Set</th>
<th>$\xi$</th>
<th>$N_{c_{max}}^{c_{max}}$</th>
<th>$N_{s_{max}}^{s_{max}}$</th>
<th>$\omega$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A–D</td>
<td>1 min{$k+2,4$}</td>
<td>4</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>1 min{$k+1,3$}</td>
<td>3 2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Results and Discussion

In the following, we present and discuss the results from the computational study, which are shown in Tables 7.5–7.9. For all instances and solution methods, we set a limit of 3600 s on the solution time. Note that the solution time does not include the time required for pre-generating the candidate routes in the exact method and in Heuristic H2 because route generation in these two methods is considered to be an offline step that is only performed once. It should be mentioned that for the larger instances, this route generation procedure takes several hours. In contrast, dynamic route generation is performed online in Heuristic H3; hence, the required time is included in the reported solution time.

The tables list the following statistics:

- $TC$ - total cost in $.
- $ST$ - solution time in s; $ST$ is not reported if the limit of 3600 s is reached.
- $OG$ - optimality gap in %, which is reported for the exact method and for Heuristic H2 if the MILP cannot be solved to zero optimality gap within the time limit; note that $OG$ is the optimality gap output by the MILP solver, it is not the gap to the true optimal solution.
7. Multiscale Production Routing in Power-intensive Supply Chains

- $NR$ - number of routes considered, reported for the exact method and Heuristic H2; recall that $NR = \sum_{t \in T} \sum_{p} \sum_{t \in T} |S_{tp}|$.

- $NR^*$ - maximum number of routes considered in an iteration of Heuristic H3.

- $NI$ - number of iterations used in Heuristic H3.

- $RD$ - relative difference to optimal (or near-optimal) solution in $\%$, i.e. $RD = (TC - \overline{TC})/\overline{TC}$ with $\overline{TC}$ being the total cost obtained from solving the MPRP exactly.

- $RI$ - relative improvement to Heuristic H1 in $\%$, i.e. $RI = (TC - TC)/TC$ with $TC$ being the total cost obtained from Heuristic H1.

Moreover, for every instance, the lowest total cost obtained from a heuristic method is shown in bold.

**Table 7.5:** Comparison of solutions for all instances in Set A.

<table>
<thead>
<tr>
<th></th>
<th>Exact</th>
<th>Heuristic H1</th>
<th>Heuristic H2a</th>
<th>Heuristic H2b</th>
<th>Heuristic H3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$TC$</td>
<td>$ST$</td>
<td>$NR$</td>
<td>$TC$</td>
<td>$RD$</td>
</tr>
<tr>
<td>A1</td>
<td>51,960</td>
<td>31</td>
<td>1797</td>
<td>61,781</td>
<td>18.9</td>
</tr>
<tr>
<td>A2</td>
<td>52,923</td>
<td>16</td>
<td>1227</td>
<td>64,026</td>
<td>21.0</td>
</tr>
<tr>
<td>A3</td>
<td>49,354</td>
<td>25</td>
<td>2157</td>
<td>60,033</td>
<td>21.2</td>
</tr>
<tr>
<td>A4</td>
<td>48,156</td>
<td>12</td>
<td>2139</td>
<td>57,412</td>
<td>19.2</td>
</tr>
<tr>
<td>A5</td>
<td>50,405</td>
<td>9</td>
<td>2892</td>
<td>61,078</td>
<td>21.2</td>
</tr>
<tr>
<td>A6</td>
<td>53,045</td>
<td>5</td>
<td>1983</td>
<td>63,555</td>
<td>19.8</td>
</tr>
<tr>
<td>A7</td>
<td>53,126</td>
<td>65</td>
<td>1386</td>
<td>67,586</td>
<td>27.2</td>
</tr>
<tr>
<td>A8</td>
<td>49,862</td>
<td>32</td>
<td>1863</td>
<td>59,722</td>
<td>19.8</td>
</tr>
<tr>
<td>A9</td>
<td>52,000</td>
<td>51</td>
<td>1470</td>
<td>59,722</td>
<td>19.8</td>
</tr>
<tr>
<td>A10</td>
<td>50,341</td>
<td>30</td>
<td>1575</td>
<td>59,468</td>
<td>18.1</td>
</tr>
<tr>
<td>Avg.</td>
<td>51,135</td>
<td>28</td>
<td>1849</td>
<td>61,868</td>
<td>21.0</td>
</tr>
</tbody>
</table>

**Table 7.6:** Comparison of solutions for all instances in Set B.

<table>
<thead>
<tr>
<th></th>
<th>Exact</th>
<th>Heuristic H1</th>
<th>Heuristic H2a</th>
<th>Heuristic H3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$TC$</td>
<td>$OG$</td>
<td>$NR$</td>
<td>$TC$</td>
</tr>
<tr>
<td>B1</td>
<td>151,685</td>
<td>0.8</td>
<td>5990</td>
<td>170,619</td>
</tr>
<tr>
<td>B2</td>
<td>155,942</td>
<td>1.5</td>
<td>4090</td>
<td>160,944</td>
</tr>
<tr>
<td>B3</td>
<td>149,268</td>
<td>1.5</td>
<td>7190</td>
<td>174,001</td>
</tr>
<tr>
<td>B4</td>
<td>143,719</td>
<td>1.7</td>
<td>7130</td>
<td>163,503</td>
</tr>
<tr>
<td>B5</td>
<td>149,232</td>
<td>1.1</td>
<td>9640</td>
<td>173,951</td>
</tr>
<tr>
<td>B6</td>
<td>157,077</td>
<td>0.7</td>
<td>6610</td>
<td>183,503</td>
</tr>
<tr>
<td>B7</td>
<td>153,999</td>
<td>1.6</td>
<td>4620</td>
<td>179,971</td>
</tr>
<tr>
<td>B8</td>
<td>143,700</td>
<td>0.9</td>
<td>6210</td>
<td>165,127</td>
</tr>
<tr>
<td>B9</td>
<td>151,021</td>
<td>1.2</td>
<td>4900</td>
<td>172,087</td>
</tr>
<tr>
<td>B10</td>
<td>146,996</td>
<td>1.4</td>
<td>5250</td>
<td>171,743</td>
</tr>
<tr>
<td>Avg.</td>
<td>150,264</td>
<td>1.3</td>
<td>6163</td>
<td>173,546</td>
</tr>
</tbody>
</table>

213
Table 7.7: Comparison of solutions for all instances in Set C.

<table>
<thead>
<tr>
<th></th>
<th>Heuristic H1</th>
<th>Heuristic H2a</th>
<th>Heuristic H2b</th>
<th>Heuristic H3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TC ST TC RI OG NR</td>
<td>TC RI OG NR</td>
<td>TC RI ST NI NR*</td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>265,473 26 228,752 13.8 0.9 5800</td>
<td>227,257 14.4 1.0 9470</td>
<td>233,491 12.0 85 5 1940</td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td>255,050 28 232,683 8.8 1.0 5650</td>
<td>230,355 9.7 1.0 8570</td>
<td>226,333 11.3 86 5 2010</td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>259,630 17 230,963 8.7 0.9 5210</td>
<td>228,874 9.5 1.0 8380</td>
<td>222,141 12.2 147 6 2050</td>
<td></td>
</tr>
<tr>
<td>C4</td>
<td>257,728 18 231,888 10.0 0.7 5860</td>
<td>229,373 11.0 1.0 9510</td>
<td>227,032 11.9 91 6 2050</td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td>260,528 56 228,780 12.2 0.7 5990</td>
<td>225,787 13.3 1.1 10,030</td>
<td>228,270 12.4 119 5 1900</td>
<td></td>
</tr>
<tr>
<td>C6</td>
<td>268,916 23 233,155 13.3 0.9 5900</td>
<td>231,582 13.9 1.2 10,010</td>
<td>235,908 12.3 109 11 1920</td>
<td></td>
</tr>
<tr>
<td>C7</td>
<td>257,249 19 239,763 6.8 0.7 5240</td>
<td>237,943 7.5 0.8 9170</td>
<td>232,596 9.6 85 7 2020</td>
<td></td>
</tr>
<tr>
<td>C8</td>
<td>249,358 78 234,768 5.9 0.5 5480</td>
<td>233,398 6.4 0.8 9220</td>
<td>220,856 11.4 94 6 2020</td>
<td></td>
</tr>
<tr>
<td>Avg.</td>
<td>260,059 31 234,373 9.8 0.8 5575</td>
<td>232,173 10.7 1.0 9298</td>
<td>230,145 11.5 109 6 1987</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.8: Comparison of solutions for all instances in Set D.

<table>
<thead>
<tr>
<th></th>
<th>Heuristic H1</th>
<th>Heuristic H2a</th>
<th>Heuristic H2b</th>
<th>Heuristic H3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TC ST TC RI OG NR</td>
<td>TC RI OG NR</td>
<td>TC RI ST NI NR*</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td>1,257,807 788 970,109 22.9 2.9 17,400</td>
<td>972,943 22.6 3.6 28410</td>
<td>952,830 24.2 2.9 9058</td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td>1,300,298 1421 1,035,147 20.4 3.9 16,950</td>
<td>1,041,365 19.9 5.0 28,290</td>
<td>991,955 23.7 4.0 9505</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>1,308,509 1285 1,030,173 21.3 3.9 16,290</td>
<td>1,037,219 20.7 5.7 27,120</td>
<td>993,238 24.1 4.1 9295</td>
<td></td>
</tr>
<tr>
<td>D4</td>
<td>1,321,906 1512 1,070,040 19.1 2.8 15,570</td>
<td>1,070,243 19.0 3.0 26,160</td>
<td>1,027,871 24.2 5 8194</td>
<td></td>
</tr>
<tr>
<td>D5</td>
<td>1,211,123 497 995,681 17.8 2.7 17,580</td>
<td>991,704 18.1 2.9 28,530</td>
<td>976,056 19.4 3.5 8459</td>
<td></td>
</tr>
<tr>
<td>D6</td>
<td>1,210,258 683 1,004,152 17.0 3.5 17,970</td>
<td>1,026,566 17.3 3.4 27,510</td>
<td>983,653 20.7 5 8459</td>
<td></td>
</tr>
<tr>
<td>Avg.</td>
<td>1,253,177 841 1,017,269 18.8 3.2 16725</td>
<td>1,015,513 18.9 3.8 27,894</td>
<td>979,557 21.8 5 8944</td>
<td></td>
</tr>
</tbody>
</table>

Table 7.9: Comparison of solutions for all instances in Set E. The average values for Heuristic H2 are computed over the available numbers.

<table>
<thead>
<tr>
<th></th>
<th>Heuristic H1</th>
<th>Heuristic H2</th>
<th>Heuristic H3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TC ST TC RI OG NR</td>
<td>TC RI OG NR</td>
<td>TC RI ST NI NR*</td>
</tr>
<tr>
<td>E1</td>
<td>1,246,916 1006 1,004,541 19.4 12.5 56,070</td>
<td>972,943 22.6 3.6 28410</td>
<td>952,830 24.2 2.9 9058</td>
</tr>
<tr>
<td>E2</td>
<td>1,231,859 908 929,386 24.6 7.2 56,520</td>
<td>1,041,365 19.9 5.0 28,290</td>
<td>991,955 23.7 4.0 9505</td>
</tr>
<tr>
<td>E3</td>
<td>1,287,454 1183 1,735,108 -34.8 49.4 58,260</td>
<td>1,037,219 20.7 5.7 27,120</td>
<td>993,238 24.1 4.1 9295</td>
</tr>
<tr>
<td>E4</td>
<td>1,250,251 497 995,681 17.8 2.7 17,580</td>
<td>991,704 18.1 2.9 28,530</td>
<td>976,056 19.4 3.5 8459</td>
</tr>
<tr>
<td>E5</td>
<td>1,210,258 683 1,004,152 17.0 3.5 17,970</td>
<td>1,026,566 17.3 3.4 27,510</td>
<td>983,653 20.7 5 8459</td>
</tr>
<tr>
<td>E6</td>
<td>1,222,821 854 1,619,761 -32.5 46.2 57,210</td>
<td>1,042,662 18.2 3.3 30,030</td>
<td>1,031,971 19.1 5 8194</td>
</tr>
<tr>
<td>E7</td>
<td>1,240,704 578 1,035,173 16.6 3.3 15,720</td>
<td>1,026,566 17.3 3.4 27,510</td>
<td>983,521 20.7 5 8730</td>
</tr>
<tr>
<td>E8</td>
<td>1,231,443 980 n/a n/a n/a</td>
<td>985,583 14.2 3.5 27,660</td>
<td>924,796 19.5 4 8993</td>
</tr>
<tr>
<td>Avg.</td>
<td>1,235,574 957 1,265,145 -2.4 26.0 58,494</td>
<td>929,615 24.7 5 11,891</td>
<td>979,557 21.8 5 8944</td>
</tr>
</tbody>
</table>

214
All instances in Set A (see Table 7.5) are solved to optimality, most of them within one minute due to the moderate number of feasible routes (on average 1849). On average, the total cost obtained with Heuristic H1 is 21% higher than the optimal total cost. Compared with Heuristic H1, Heuristics H2a, H2b, and H3 achieve significantly improved solutions. Heuristics H2a and H3 provide solutions of similar quality, on average within 2.5% to optimality. As expected, Heuristic H2b outperforms Heuristic H2a since it considers additional routes; in fact, for all 10 instances, the best heuristic solutions are obtained with Heuristic H2b.

Unlike in Set A, the instances in Set B are not solved to optimality within the given time limit; however, near-optimal solutions are obtained, where the optimality gap is on average 1.3%. Also the MILPs used in Heuristics H2a and H2b are solved with nonzero optimality gaps; however, the obtained solutions are close to optimal, some even better than the ones obtained with the exact method (indicated by a negative $RD$). Here, one can observe that a solution obtained with Heuristic H2b may not be as good as the one obtained with Heuristic H2a because the MILPs are not solved to optimality. Heuristic H3 again achieves high-quality solutions, but does not perform as well as Heuristics H2a and H2b.

Solving the MPRP exactly becomes computationally intractable for instances in Sets C–E; hence, we only show results from the heuristic algorithms in Tables 7.7–7.9. Note that here we compare the results with the solutions obtained with Heuristic H1, and $RI$ is defined such that the larger $RI$, the better the solution. With increasing problem size, the MILPs considered in Heuristic H2 become more difficult to solve, resulting in reduced solution quality. This effect is less pronounced in Heuristic H3 because of its dynamic route generation procedure that keeps the route set sufficiently small. As a result, in Set C, the best solutions to 7 of the 10 instances are obtained with Heuristic H3. In Sets D and E, Heuristic H3 consistently achieves the best solution.

From the results for Set E (see Table 7.9), one can see that the performance of Heuristic H2 deteriorates in these large instances. Due to the large number of candidate routes, solving the MILPs in Heuristic H2 becomes intractable. In three instances, the optimality gaps obtained after one hour are still close to 50%; in three other instances, where no numerical results are reported (n/a), the solver was not able to find any feasible solutions within the time limit. Heuristic H3, however, still achieves good feasible solutions with significantly lower costs than the ones obtained by Heuristics H1 and H2.

In summary, among all solution methods, Heuristic H1 exhibits the worst per-
formance in terms of solution quality due to the inaccurate representation of the
distribution constraints, which results in inefficient routing decisions and large
additional product purchases in Phase 2. Heuristic H2 performs well in small in-
stances, where one can afford generating a sufficiently large number of routes to
obtain good solutions; however, the performance deteriorates in larger instances.
In contrast, the proposed solution method, Heuristic H3, consistently obtains high-
quality solutions in a few iterations and significantly outperforms the other solu-
tion methods in the larger instances.

7.5.3. Industrial Case Study

We now apply the proposed MPRP framework to a real-world industrial test case
provided by Praxair. Here, we consider an industrial gas business that produces
and sells liquid oxygen (LO2), liquid nitrogen (LN2), gaseous oxygen (GO2), and
gaseous nitrogen (GN2). While LO2 and LN2 can be stored and transported to
customer sites using tractor-trailers, GO2 and GN2 are nonstorable and have to be
distributed via pipelines immediately after their production; hence, routing deci-
sions only involve liquid product customers. We consider a supply chain consisting
of 2 plants, P1 and P2, and approximately 240 customers. The two plants have a
combined fleet of 10 LO2 and 10 LN2 tractor-trailers. While Plant P1 has to satisfy
demand for both liquid and gaseous products, Plant P2 only serves liquid product
customers.

The production process, namely cryogenic air separation, is highly power-
intensive such that the vast majority of the variable production cost is the cost of
electricity. Electricity prices can vary significantly across different locations. In this
case, Plant P1 participates in the day-ahead market in which the price varies over
time, whereas Plant P2 purchases power at a constant unit price. A forecast of the
day-ahead prices is available for the given planning horizon.

The MPRP is solved for a planning horizon of 4 weeks, where we choose \( \Delta t^f \)
and \( \Delta t^c \) to be 4 h and 12 h, respectively, resulting in 168 level-1 and 56 level-2 time
periods. We apply the proposed algorithm to this large-scale MPRP and present
the solution obtained after one hour runtime.

Figure 7.9 shows the electricity consumption and price profiles for both plants
over the entire planning horizon. One can see that the electricity price at Plant P2
is significantly higher than the average electricity price at Plant P1. As a result, in
order to reduce energy cost, Plant P2 is shut down three times for extensive periods
of time and also at the end of the planning horizon. One can further see that the
solution suggests load shifting at Plant P1 in order to take advantage of low-price hours.

![Electricity consumption and electricity price profiles for each plant.](image)

**Figure 7.9:** Electricity consumption and electricity price profiles for each plant.

There is a trade-off between production and distribution costs that is not apparent from Figure 7.9. Although the electricity price is almost always lower at Plant P1, it does not utilize its full production capacity, i.e. more production could be shifted from Plant P2 to Plant P1. However, the higher production cost is offset by the reduction in distribution cost because more customers are located closer to Plant P2 than to Plant P1.

Figures 7.10 and 7.11 show the product flows and inventory profiles for the liquid products at Plants P1 and P2, respectively. In Figure 7.10, one can clearly see the effect of load shifting at Plant P1. At Plant P2, inventory is accumulated during hours of production such that products can be drawn from the inventory and distributed to the customers when the plant is shut down, as depicted in Figure 7.11.

Now we compare our solution with the ones obtained from two alternative solution methods. The first method applies a similar approach as Heuristic H1, however, with more sophisticated and tailored constraints on the distribution resources. In the following, we refer to this approach as Heuristic PH1. The second approach is an extension of Heuristic PH1, referred to as Heuristic PH2, which further incorporates fixed costs for customer visits. The fixed distribution costs in Heuristic PH2 prevent the model from suggesting a large number of deliveries with small quantities; however, they also introduce additional binary variables that considerably increase the computational complexity.

Heuristics PH1, PH2, and H3, with the latter being our proposed algorithm with dynamic route generation, apply equivalent representations of the production side; however, the distribution side is modeled with different levels of accuracy.
For this comparative study, we first apply Heuristics PH1, PH2, and H3 to obtain the production plan and the plant-to-customer allocation decisions for each of the three solution approaches. Then, the same routing tool is applied to the three sets of plant-to-customer allocation decisions to determine optimal (or near-optimal) routes and accurate routing costs.

Table 7.10 compares the solutions obtained from Heuristics PH1, PH2, and H3. The table shows the breakdown of the total costs ($TC$) into the production costs ($CPD$) and distribution costs ($CDI$) for each plant. In this test case, no additional product purchase is required, and inventory costs are negligible; hence, these costs are omitted. Furthermore, the table shows the computation time for each solution method. In terms of total cost, Heuristic H3 outperforms both Heuristics PH1 and PH2, with relative cost savings of 8.7 and 2.4%, respectively, which can be attributed to the more rigorous modeling of routing decisions. One can see that compared to Heuristics PH1 and PH2, Heuristic H3 suggests producing less at Plant P1 and more at Plant P2. This production plan results in higher total production cost, but in overall proves to be the better choice since the routing cost can be significantly reduced by distributing more from Plant P2.
7. Multiscale Production Routing in Power-intensive Supply Chains

Figure 7.11: Production quantities, shipments, and inventory levels of LO2 and LN2 at Plant P2.

Table 7.10: Comparison of costs and solution times for the industrial test case.

<table>
<thead>
<tr>
<th></th>
<th>Heuristic PH1</th>
<th>Heuristic PH2</th>
<th>Heuristic H3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TC</strong></td>
<td>100.00</td>
<td>93.46</td>
<td>91.26</td>
</tr>
<tr>
<td><strong>C(P)D_{P1}</strong></td>
<td>32.67</td>
<td>32.66</td>
<td>31.88</td>
</tr>
<tr>
<td><strong>C(P)D_{P2}</strong></td>
<td>13.05</td>
<td>13.12</td>
<td>15.01</td>
</tr>
<tr>
<td><strong>C(D)I_{P1}</strong></td>
<td>42.53</td>
<td>36.61</td>
<td>32.40</td>
</tr>
<tr>
<td><strong>C(D)I_{P2}</strong></td>
<td>11.75</td>
<td>11.07</td>
<td>11.97</td>
</tr>
<tr>
<td><strong>ST [s]</strong></td>
<td>218</td>
<td>900</td>
<td>3600</td>
</tr>
</tbody>
</table>

Figure 7.12 shows for each day of the planning horizon the number of customers to visit as suggested by each of the three solutions. While Heuristic PH1 proposes to visit on average 66 customers per day, the average numbers of visited customers per day are 30 and 25 for Heuristics PH2 and H3, respectively. Heuristic PH1 creates many deliveries with small quantities, which leads to inefficient routes. This effect is mitigated in Heuristic PH2 by introducing fixed distribution costs, ultimately resulting in lower routing costs. However, the improved solution quality comes at the cost of higher computational expense. While Heuristic PH1 solves in 218 s, the solution from Heuristic PH2 is obtained after 900 s. Among the
three solution approaches, Heuristic H3 obtains the best solution, but only after 3600 s.

Figure 7.12: Comparison of the numbers of customers to be visited on each day of the planning horizon as suggested by Heuristics PH1, PH2, and H3.

In practice, under normal circumstances, the plant-to-customer allocation is fixed, i.e. each customer is assigned to a particular plant and only receives delivery from this plant, which may limit the flexibility in the supply chain operations. To compare the differences between the proposed solutions and the current practice, we show in Figure 7.13 for each of the three solutions the changes in plant-to-customer allocation compared to the current plant-to-customer allocation. Here, an allocation change is defined as one customer that is to be visited in the corresponding solution from a plant different from the one to which it is currently assigned. The number of allocation changes can be interpreted as a measure for the amount of disruption in the default assignment required to obtain the suggested solution. In practice, small changes are desired; a large number of allocation changes may suggest that the current plant-to-customer allocation or the current assignment of vehicles to plants is inadequate. In this case, significantly fewer allocation changes, on average 7 per day, are required for Heuristic H3 than for Heuristics PH1 and PH2, which require on average 24 and 11 allocation changes per day, respectively.

Another advantage of Heuristic H3 is that it only considers feasible routes; hence, the proposed deliveries are guaranteed to be feasible. In contrast, Heuristics PH1 and PH2 may make plant-to-customer allocation decisions that are infeasible in the subsequent routing step, in the sense that not all proposed deliveries can actually be made. In this particular test case, routing infeasibility does not occur because the customers are located relatively close to each other such that the limit on the travel distance is not an issue. However, in other supply chain networks with longer inter-customer distances, the situation of routing infeasibility may very well
7. Multiscale Production Routing in Power-intensive Supply Chains

Figure 7.13: Comparison of the numbers of plant-to-customer allocation changes from the current assignment required for Heuristics PH1, PH2, and H3.

arise when Heuristics PH1 and PH2 are applied.

7.6. Summary

In this chapter, we have introduced the multiscale production routing problem, which considers the integrated optimization of production, inventory, distribution, and routing decisions in multicommodity supply chains with complex production facilities. In the MPRP, the objective is to make decisions at two different levels: operational scheduling decisions on the production side and tactical plant-to-customer allocation decisions on the distribution side.

The proposed MILP model incorporates two different time scales. For production scheduling, a mode-based formulation is applied to the fine time grid such that all critical operational features, including interdependent production rates, limitations on transitions between operating points, and time-sensitive production costs, can be captured. In addition, for distribution planning, vehicle routing is considered in each time period of the coarse time grid. An iterative heuristic solution method has been developed in order to solve large instances of the MPRP. At each iteration of the proposed algorithm, a restricted MPRP considering a subset of all possible routes is solved, and the set of candidate routes is updated based on the solutions obtained in previous iterations.

The proposed MPRP framework has been applied to an illustrative example, in a computational study with 50 instances of various sizes, as well as to an industrial test case with real-world data provided by Praxair. In the computational study, where the largest instances consider supply chains with 3 products, 3 plants, and 300 customers, the proposed algorithm is compared with a standard two-phase heuristic and a solution strategy involving a one-time heuristic pre-generation of
candidate routes. The results show that the proposed algorithm finds high-quality solutions in reasonable computation times and significantly outperforms the other two solution approaches in large instances. In the industrial case study, which considers a real-world industrial gas supply chain with 2 plants and approximately 240 customers and a planning horizon of 4 weeks, the proposed algorithm outperforms available alternative solution approaches in terms of solution quality, although longer computation times are required.
8. On the Relation Between Flexibility Analysis and Robust Optimization for Linear Systems

In Chapters 5 and 6, we have applied robust optimization to solve optimization problems under uncertainty. Flexibility analysis is another approach to modeling decision-making under uncertainty that has been developed in the PSE community, and it turns out that flexibility analysis and robust optimization share some fundamental concepts, such as the use of polyhedral uncertainty sets and the worst-case approach to guarantee feasibility. However, the connection between these two approaches has not been sufficiently acknowledged and examined in the literature. In this chapter, we present a comparison between flexibility analysis and robust optimization from a historical perspective, and further establish the link between these two approaches by proposing new formulations for the three classical flexibility analysis problems—flexibility test, flexibility index, and design under uncertainty—based on duality theory and the affinely adjustable robust optimization approach.

This chapter is organized as follows. After the work is further motivated in Section 8.1, Section 8.2 presents a historical perspective on flexibility analysis and robust optimization, which have been developed independently from each other in different research communities. In Sections 8.3–8.5, the three flexibility analysis problems are introduced, and for each of them, the formulations resulting from three different solution approaches are presented. These formulations are then applied to three numerical examples in Section 8.6. In Section 8.7, we close with a summary of the main results.
8.1. Background

Process flexibility and resiliency analysis of chemical processes using mathematical optimization models have received significant attention in the PSE community for more than thirty years. In general, the study of the flexibility of a process can be addressed at two stages: (1) at the design stage, where the optimization models for process design explicitly incorporate flexibility constraints, or (2) when for a fixed design, there is the need to evaluate the process flexibility in the presence of variations of some operating conditions, which at the design stage were considered constant, but in reality are subject to uncertainty.

In a recent review paper, Grossmann et al. (2014) give a historical perspective on the evolution of the concepts and mathematical models used for flexibility analysis. Some of the earlier works in this area include the work of Friedman & Reklaitis (1975a,b) from the mid 70s, who proposed techniques for solving LPs with some of the parameters subject to uncertainty; subsequently from the 80s the resiliency concepts proposed by Saboo et al. (1985) and applied to heat exchanger networks; and the development of optimization models to quantify process flexibility by solving the flexibility test and flexibility index problems (Halemane & Grossmann, 1983; Swaney & Grossmann, 1985a).

The analysis of the flexibility of chemical processes is closely related to the solution of optimization problems under uncertainty. In the works mentioned above, one assumes that each uncertain parameter can take any value within a given range, and the process is considered sufficiently flexible if feasible operation can be achieved for the entire parameter range. Interestingly, the exact same idea forms the foundation of robust optimization (Ben-Tal et al., 2009), which was introduced in the late 90s. The increasing interest in robust optimization in recent years has led to many theoretical developments and its application to a wide range of problems (Bertsimas et al., 2011; Gabrel et al., 2014).

The two main concepts shared by flexibility analysis and robust optimization are the following:

- using polyhedral uncertainty sets to describe parameter uncertainty;
- applying worst-case analysis to test feasibility.

However, while the most effective solution approaches developed in flexibility analysis make use of the Karush-Kuhn-Tucker (KKT) conditions and insights on the possible sets of active constraints as it is aimed at solving nonlinear models, ro-
Robust optimization applies duality theory to mostly linear models to obtain tractable formulations.

Another major difference between flexibility analysis and “traditional” robust optimization is the treatment of recourse (reactive actions after the realization of the uncertainty). While flexibility analysis explicitly considers control variables that can be adjusted depending on the realized values of the uncertain parameters, robust optimization traditionally does not account for recourse, which often leads to overly conservative solutions. This gap between the two approaches is being bridged by the recent development of the adjustable robust optimization concept (Ben-Tal et al., 2004; Bertsimas & Goyal, 2010; Kuhn et al., 2011), which allows the incorporation of recourse decisions, often in the form of linear decision rules.

In this work, we consider a set of linear inequality constraints with a given general structure for which the following three flexibility analysis problems are addressed:

1. the flexibility test problem;
2. the flexibility index problem;
3. design under uncertainty with flexibility constraints.

For linear models described by inequalities, we establish the link between flexibility analysis and robust optimization by first developing a new flexibility analysis approach based on duality. Then, we apply the affinely adjustable robust optimization (Ben-Tal et al., 2004) approach to the same flexibility analysis problems. Hence, we present and compare the following three approaches:

1. traditional (KKT-based) flexibility analysis (TFA);
2. duality-based flexibility analysis (DFA);
3. affinely adjustable robust optimization (AARO).

While the TFA and DFA approaches lead to MILP formulations, only LPs need to be solved in the AARO approach. However, we show that while the TFA and DFA approaches obtain the same optimal solution, the AARO approach may predict a lower level of flexibility due to the restriction of the recourse to linear functions of the uncertain parameters.
8. Relation Between Flexibility Analysis and Robust Optimization

8.2. Historical Perspective

While flexibility analysis was developed in the PSE community, robust optimization is recognized as a subfield of operations research (OR). In the following, we take a look at the historical development of these two research areas, which interestingly have evolved almost entirely independently from each other.

In their seminal work from 1975, Friedman & Reklaitis (1975a,b) address the problem of solving LPs with possibly correlated uncertain parameters, where each uncertain parameter can take any value within a known range of variation. A worst-case approach is proposed which aims at finding a solution that is feasible for any possible realization of the uncertainty. General nonlinear systems are considered in the work by Grossmann & Sargent (1978), which forms the basis for all later contributions in the area of flexibility analysis. The major conceptual innovation is the distinction between design and control variables; while design variables are chosen at the design stage and cannot be changed during the operation of the plant, control variables can be adjusted depending on the realization of the uncertain parameters. This concept resembles the stage-wise construction of stochastic programming (Birge & Louveaux, 2011) models and realistically represents the decision-making process in chemical plant design and operation.

Most of the later theoretical work in flexibility analysis was conducted in the 1980s by Grossmann and coworkers. Halemane & Grossmann (1983) introduce a rigorous formulation of the problem of design under uncertainty. A max-min-max constraint, which involves solving what is known as the flexibility test problem, guarantees the existence of a feasible region for the specified range of parameter values. One main result is that if the constraints are convex, the solution of the flexibility test problem lies at a vertex of the polyhedral region of parameters. Based on this insight, a vertex enumeration formulation and an iterative cutting plane algorithm are proposed to solve the design under uncertainty problem. Swaney & Grossmann (1985a,b) introduce the flexibility index problem by proposing a quantitative index which measures the size of the parameter space over which feasible operation can be attained. Two algorithms have been proposed that are designed to avoid exhaustive vertex enumeration. Realizing that the flexibility test and flexibility index problems result in bilevel formulations, Grossmann & Floudas (1987) replace the lower-level problems by their KKT conditions and apply an active-constraint strategy to convert the bilevel problems into single-level mixed-integer problems. The derivation of the model does not require the assumption of vertex
solutions; hence, it is able to predict nonvertex critical points.

Pistikopoulos & Grossmann (1989a,b) consider the optimal retrofit design with the objective of improving process flexibility. Further extensions include stochastic flexibility (Pistikopoulos & Mazzuchi, 1990; Straub & Grossmann, 1990), where the uncertain parameters are described by a joint probability distribution function; flexibility analysis of dynamic systems (Dimitriadis & Pistikopoulos, 1995); flexible design with confidence intervals and process variability (Rooney & Biegler, 2001, 2003); new flexibility measure from constructing feasible polytopes in the parameter space (Ierapetritou, 2001); and simplicial approximation of feasibility limits (Goyal & Ierapetritou, 2002, 2003). More recent works focus on data-driven approaches for flexibility analysis (Banerjee & Ierapetritou, 2005; Banerjee et al., 2010; Boukouvala & Ierapetritou, 2012).

While the area of flexibility analysis has evolved over the last 40 years, the history of robust optimization is very different. The work that is generally recognized as the first contribution to the area of robust optimization is a short technical note from 1973 by Soyster (1973), who considers robust solutions to LPs with columnwise uncertainty, which can be seen as a special case of the problem addressed by Friedman & Reklaitis (1975a,b). Then, essentially no further development was made in the OR community until Soyster’s work was rediscovered in 1998 by Ben-Tal & Nemirovski (1998) and El Ghaoui et al. (1998), who introduced the notion of uncertainty sets and robust counterparts. While an uncertainty set is the set of all possible realizations of the uncertainty, a robust counterpart is a formulation that constrains the original model equations to be feasible for every possible realization of the uncertainty.

The major concern in robust optimization is computational tractability, which strongly depends on the structure of the optimization problem and the form of the uncertainty set. Tractable robust counterparts have been derived for a wide variety of optimization problems (Ben-Tal et al., 2009), often by using duality theory. For example, the robust counterpart of an LP with uncertain parameters defined by polyhedral uncertainty sets can be formulated as an LP of similar complexity (Ben-Tal & Nemirovski, 1999); the robust counterpart of an LP with ellipsoidal uncertainty can be posed as a second-order cone program (Ben-Tal & Nemirovski, 1999); and for quadratically constrained quadratic programs with simple ellipsoidal uncertainty, the robust counterpart is a semidefinite program (Ben-Tal et al., 2002). Some of the results for continuous problems can be extended to models with discrete variables (Bertsimas & Sim, 2003). For many other classes of optimiza-
tion problems, however, only tractable approximate solution approaches have been proposed so far (El Ghaoui et al., 1998; Ben-Tal & Nemirovski, 2002; Bertsimas & Sim, 2006). In particular, obtaining robust solutions for general nonlinear systems remains a major challenge.

One drawback of robust optimization is that the result may often be overly conservative since the approach optimizes for the worst case. To address this issue, Bertsimas & Sim (2004) introduce the notion of a budget of uncertainty, which can be changed in order to adjust the level of conservatism in the solution. In the proposed approach, the budget of uncertainty is encoded in the form of a cardinality constraint on the number of uncertain parameters that are allowed to deviate from their nominal values, and probabilistic bounds for constraint satisfaction as functions of the budget parameter are derived. The concept of probabilistic guarantees in robust optimization traces back to Ben-Tal & Nemirovski (2000), who show that for an ellipsoid with radius \( \Omega \), the corresponding robust feasible solution satisfies the original constraint with probability at least \( 1 - e^{-\Omega^2/2} \). Since then, probabilistic guarantees have been derived for a number of different robust formulations (Bertsimas et al., 2004; Bertsimas & Sim, 2006; Chen et al., 2007).

Unlike flexibility analysis, traditional static robust optimization does not consider recourse, which is another limitation that may lead to overly conservative solutions. A more realistic approach has been introduced with the concept of adjustable/adaptable robust optimization (Ben-Tal et al., 2004; Bertsimas & Goyal, 2010), which involves adjustable recourse decisions that depend on the realization of the uncertainty. Benders-dual cutting plane (Bertsimas et al., 2013) and column-and-constraint generation (Zhao & Zeng, 2012; Zeng & Zhao, 2013) algorithms have been proposed to solve the two-stage robust optimization problem with fully adjustable recourse. However, solving fully adjustable robust optimization problems is very difficult and in many cases intractable; an effective way to reduce the computational effort is to describe the recourse decisions as functions of the uncertain parameters and then restrict these functions to specific tractable forms. Several classes of recourse functions have been proposed in the literature (Bemporad et al., 2003; Chen & Zhang, 2009b; Bertsimas et al., 2011), among which the affine or linear decision rules (Ben-Tal et al., 2004) are most popular. While affine decision rules allow the formulation of efficient multistage models and are shown to be optimal for some specific types of problems (Bertsimas et al., 2010; Bertsimas & Goyal, 2012), they generally form a restriction of the fully adjustable case and hence are suboptimal. Kuhn et al. (2011) estimate the approximation error
from using linear decision rules by applying them to both the primal and the dual of the problem.

More recent developments in robust optimization include adjustable robust optimization with integer recourse (Bertsimas & Georghiou, 2015; Postek & den Hertog, 2014; Hanasusanto et al., 2015) and distributionally robust optimization (Delage & Ye, 2010; Goh & Sim, 2010), which addresses the problem of ambiguity in the probability distribution that describes the uncertain parameters. We refer to recent review papers (Bertsimas et al., 2011; Gabrel et al., 2014) for a more comprehensive overview of the advances made by the OR community in this rapidly growing research area.

In recent years, robust optimization has also been applied by the PSE community, mainly to production scheduling problems (Lin et al., 2004; Janak et al., 2007; Li & Ierapetritou, 2008b; Verderame & Floudas, 2009; Mitra et al., 2012b; Vujanic et al., 2012; Amaran et al., 2016) involving uncertainty in prices, product demands, processing times, etc. Most works apply the static robust optimization approach without recourse. Gounaris et al. (2013) derive robust counterparts for the capacitated vehicle routing problem. Li et al. (2011) apply robust optimization to linear and mixed-integer linear optimization problems, and present a systematic study of the robust counterparts for various uncertainty sets and their geometric relationships. In a subsequent work, Li et al. (2012) derive probabilistic guarantees for constraint satisfaction when applying these robust counterparts.

In Figure 8.1, we illustrate the evolution of flexibility analysis and robust optimization over time by showing a timeline with some of the most seminal works from each research area. One can see that flexibility analysis has a much longer history than robust optimization; but interestingly, it is not mentioned in any robust optimization literature. In this work, we show the strong connections between the two approaches, and highlight the fact that some of the fundamental concepts in robust optimization have already been developed in the area of flexibility analysis long before the era of robust optimization.

Furthermore, it is interesting to note that some concepts from flexibility analysis and robust optimization have been independently developed and applied in a third field, namely the field of robust design of mechanical systems (Du & Chen, 2000; Yao et al., 2011). In particular, the so-called corner space evaluation method (Sundaresan et al., 1995) strongly resembles the vertex enumeration method from flexibility analysis.
8. Relation Between Flexibility Analysis and Robust Optimization

Figure 8.1: Timeline showing some of the seminal works in flexibility analysis (FA), robust optimization (RO), and robust optimization in PSE.

8.3. Flexibility Test Problem

8.3.1. Problem Statement

Consider a set of $m$ linear inequality constraints of the following form:

$$f_j(d, z, \theta) = a_j d + b_j z + c_j \theta \leq 0 \quad \forall j \in J$$  \hspace{1cm} (8.1)

where $d \in \mathbb{R}^{n_d}$ are design variables, $z \in \mathbb{R}^{n_z}$ are control variables, and $\theta \in \mathbb{R}^{n_\theta}$ are uncertain parameters; $J = \{1, 2, \ldots, m\}$ is the set of constraints, $a_j$, $b_j$, and $c_j$ are row vectors of appropriate dimensionalities. In engineering applications, the inequalities typically represent restrictions on the operating conditions of a process and product requirements. In a general linear model involving equality and inequality constraints, they result from eliminating the state variables from the equations and substituting them in the inequalities. Note that constant summands in the constraint functions can be incorporated in this general formulation by defining fixed dummy design variables.

The flexibility test problem (Halemane & Grossmann, 1983) can then be stated as follows: For a given design $d$, determine whether by proper adjustment of the
control variables $z$, the inequalities $f_j(d,z,\theta) \leq 0$, $j \in J$, hold for all $\theta \in T = \{\theta : \theta^L \leq \theta \leq \theta^U\}$. Here, $T$ denotes the uncertainty set that takes the form of an $n_\theta$-dimensional hyperbox.

8.3.2. Traditional Flexibility Analysis

For fixed $d$ and $\theta$, the feasibility function is defined as

$$
\psi(d,\theta) = \min_{z \in \mathbb{R}^{n_z}} \max_{f_j \in J} f_j(d,z,\theta)
$$

which returns the smallest of the largest $f_j$ that can be achieved by adjusting $z$. If $\psi(d,\theta) \leq 0$, we can have feasible operation; if $\psi(d,\theta) > 0$, the operation is infeasible regardless how we choose $z$. $\psi(d,\theta)$ can be obtained by solving the following LP:

$$
\begin{align*}
\psi(d,\theta) = \min_{z,u} & \quad u \\
\text{s.t.} & \quad A_d + B_z + C_\theta \leq u \ e \\
& \quad z \in \mathbb{R}^{n_z}, \ u \in \mathbb{R},
\end{align*}
$$

where $e$ denotes a column vector of appropriate dimensionality where all entries are 1.

The flexibility test problem is equivalent to checking whether the maximum value of $\psi(d,\theta)$ is less than or equal to zero over the entire range of $\theta$. Hence, it can be formulated as (Halemane & Grossmann, 1983)

$$
\chi(d) = \max_{\theta \in T} \psi(d,\theta) = \max_{\theta \in T} \min_{z \in \mathbb{R}^{n_z}} \max_{f_j \in J} f_j(d,z,\theta)
$$

where $\chi(d)$ corresponds to the flexibility function of design $d$ with respect to the uncertainty set $T$. The geometric interpretation of the flexibility test problem is illustrated in Figure 8.2, where for each of the two examples, the feasible region, which is a polyhedron in the $(2 + n_z)$-dimensional $(\theta,z)$-space, is projected onto the 2-dimensional $\theta$-space. In the example shown in Figure 8.2a, the problem is feasible for all $\theta \in T$ as the rectangle describing $T$ is inscribed in the projection of the feasible region, resulting in $\chi(d) \leq 0$. In fact, here, $\chi(d) = 0$ since one of the vertices of $T$ touches the boundary of the feasible region. In the second example shown in Figure 8.2b, the given design is not feasible for all $\theta \in T$ as $T$ is not completely contained in the projection of the feasible region.

An important property from flexibility analysis is expressed in the following
Theorem 8.1. If the constraint functions $f_j(d, z, \theta)$ are jointly convex in $z$ and $\theta$, Problem (FT) has its global solution at a vertex of the polyhedral region $T = \{ \theta : \theta^L \leq \theta \leq \theta^U \}$.

In the case of (8.1), the constraint functions are clearly jointly convex in $z$ and $\theta$ since they only appear linearly in $f_j$. Hence, we can apply Theorem 8.1 and solve the flexibility test problem by evaluating $\psi(d, \theta)$ at all vertices of $T$. Based on this property, Swaney & Grossmann (1985b) have proposed two solution algorithms (direct search and implicit enumeration), which search over the set of vertices, but are designed to avoid exhaustive enumeration.

The solution approach that we consider here is an active-set method (Grossmann & Floudas, 1987) that formulates the flexibility test problem as one single MILP. In this approach, the flexibility test problem given by (FT) is posed as a bilevel problem in which Problem (FF) is the lower-level problem used to compute $\psi(d, \theta)$. A single-level formulation is achieved by replacing the lower-level problem with its KKT conditions and by modeling the choice of the set of active constraints with mixed-integer constraints. The resulting flexibility test formula-
8. Relation Between Flexibility Analysis and Robust Optimization

The objective is as follows:

\[
\chi(d) = \max_{d, z, u, \lambda, s, y} u
\]

s.t. \[ \begin{align*}
A d + B z + C \theta + s & = u e \\
e^T \lambda & = 1 \\
B^T \lambda & = 0 \\
\lambda & \leq y \\
s & \leq M(e - y) \\
e^T y & \leq n_z + 1 \\
\theta & \in T, z \in \mathbb{R}^{n_z}, u \in \mathbb{R}, \lambda \in \mathbb{R}_+^m, s \in \mathbb{R}_+^m, y \in \{0, 1\}^m,
\end{align*} \] (FT_FFA)

where \( s \) is the vector of slack variables, \( \lambda \) denotes the vector of Lagrange multipliers, and \( M \) is a big-M parameter. Note that Problem (FT_FFA) has \( 3m + n_z + 2n_\theta + 2 \) constraints, \( 2m + n_z + n_\theta + 1 \) continuous variables, and \( m \) binary variables.

8.3.3. Duality-based Flexibility Analysis

We now present an alternative approach, which is derived using LP duality theory and makes use of Theorem 8.1. First, we define a new uncertainty set that only contains the vertices of \( T \):

\[
\overline{T} = \{ \theta : \theta_i = \theta_i^N + x_i \Delta \theta_i^+ - (1 - x_i) \Delta \theta_i^- \}, \quad x_i \in \{0, 1\} \quad \forall \ i \in \Theta
\] (8.3)

where \( \Delta \theta^+ = \theta^U - \theta^N, \Delta \theta^- = \theta^N - \theta^L \), and \( \Theta = \{1, 2, \ldots, n_\theta\} \) is the set of uncertain parameters. The vertices are expressed by using the binary variables \( x \). If \( x_i = 1 \), \( \theta_i = \theta_i^U \); otherwise, \( \theta_i = \theta_i^L \).

The dual of Problem (FF) is

\[
\psi(d, \theta) = \max_{\lambda} \quad (A d + C \theta)^T \lambda
\]

s.t. \[ \begin{align*}
e^T \lambda & = 1 \\
- B^T \lambda & = 0 \\
\lambda & \in \mathbb{R}_+^m,
\end{align*} \] (8.4a,b,c,d)

where \( \lambda \) is the vector of nonnegative dual variables. Due to strong duality, Problems (FF) and (8.4) achieve the same objective function value at the optimal solu-
8. Relation Between Flexibility Analysis and Robust Optimization

tion. Hence, the flexibility test problem can be reformulated as:

\[ \chi(d) = \max_{\theta \in \mathcal{T}} \max_{\lambda} (Ad + C\theta)^T \lambda \]  
\[ \text{s.t. } e^T \lambda = 1 \]  
\[ -B^T \lambda = 0 \]  
\[ \lambda \in \mathbb{R}^m_+ \]  

which is equivalent to Problem (FT) since the optimal solution is guaranteed to lie at one of the vertices of \( T \). The inner and outer maximization problems can be merged in order to achieve a single-level problem. We can then rewrite the objective function as follows:

\[ (Ad + C\theta)^T \lambda = d^TA^T \lambda + \theta^T C^T \lambda \]
\[ = d^TA^T \lambda + \left[ \theta^N + \text{diag}(x) \Delta \theta^+ - (I - \text{diag}(x)) \Delta \theta^- \right]^T C^T \lambda \]
\[ = d^TA^T \lambda + \sum_{j \in J} c_j \left[ \theta^N + \text{diag}(x) \Delta \theta^+ - (I - \text{diag}(x)) \Delta \theta^- \right] \lambda_j \]  

where \( I \) is the identity matrix of appropriate dimensionality and \( \text{diag}(\cdot) \) denotes a diagonal matrix. Note that \( C \) is an \( m \times n_\theta \) matrix, and while \( c_j \) denotes the \( j \)th row vector of \( C \), \( c_{ji} \) is the element at the \( j \)th row and \( i \)th column of \( C \).

The objective function now contains the bilinear terms \( \lambda_j x_i \). This bilinearity can be eliminated by applying exact linearization to the bilinear terms (Glover, 1975). By doing so, we obtain the following MILP reformulation of the flexibility test problem:

\[ \chi(d) = \max_{\lambda, x} \quad d^TA^T \lambda + \sum_{j \in J} \sum_{i \in \Theta} c_{ji} \left[ \lambda_j \left( \theta_i^N - \Delta \theta_i^- \right) + \bar{\lambda}_{ij} (\Delta \theta_i^+ + \Delta \theta_i^-) \right] \]
\[ \text{s.t. } e^T \lambda = 1 \]
\[ -B^T \lambda = 0 \]
\[ \bar{\lambda}_{ij} \geq (\lambda_j - 1) + x_i \quad \forall \ i \in \Theta, \ j \in J \quad (\text{FT}_{\text{DFA}}) \]
\[ \bar{\lambda}_{ij} \leq \lambda_j \quad \forall \ i \in \Theta, \ j \in J \]
\[ \bar{\lambda}_{ij} \leq x_i \quad \forall \ i \in \Theta, \ j \in J \]
\[ \lambda \in \mathbb{R}^m_+, \; \bar{\lambda} \in \mathbb{R}_+^{n_\theta \times m}, \; x \in \{0, 1\}^{n_\theta}, \]

234
which consists of \( 3m n_\theta + n_z + 1 \) constraints, \( m(n_\theta + 1) \) continuous variables, and \( n_\theta \) binary variables.

### 8.3.4. Affinely Adjustable Robust Optimization

In the following, we derive the affinely adjustable robust optimization formulation for the flexibility test problem, and show its relationship to the flexibility analysis approach. In particular, we show that the AARO approach can be overly conservative, i.e. the obtained flexibility function value may be positive although \( z \) can be adjusted such that the design is feasible for all \( \theta \in T \).

Following a general adjustable robust optimization approach (Ben-Tal et al., 2004), the flexibility test problem can be formulated as follows:

\[
\chi(d) = \min_{z(\theta), u \in \mathbb{R}} u \\
\text{s.t. } A d + B z(\theta) + C \theta \leq u e \quad \forall \theta \in T, \tag{8.7a}
\]

where \( z \) is expressed as a function of \( \theta \) since it can be seen as the vector of recourse variables that are chosen after the realization of the uncertainty. Eq. (8.7b) states that all constraints have to be satisfied for every possible realization of the uncertainty, i.e. for all \( \theta \in T \).

To show the connection with flexibility analysis, we write Problem (8.7) equivalently as

\[
\chi(d) = \min_{z(\theta)} \max_{\theta \in T} \max_{j \in J} f_j(d, z(\theta), \theta). \tag{8.8}
\]

The control function \( z(\theta) \) itself has to be prespecified and cannot be changed later on, which is why it is a decision made in the outer minimization problem. Problem (8.8) is equivalent to Problem (FT) if \( z(\theta) \) is chosen such that it is the optimal solution of Problem (FF) for every \( \theta \in T \).

### Applying Affine Control Functions

Problem (8.7) is a semi-infinite program since the number of possible control functions and the number of possible realizations of the uncertainty are both infinite. In order to derive a computationally tractable model that still guarantees feasibility for all \( \theta \in T \), we restrict the set of control functions that can be selected. In static robust optimization, \( z \) has to be a constant, which means that no recourse is
allowed. In AARO, \( z \) is expressed as an affine function of \( \theta \), i.e. \( z(\theta) = p + Q\theta \), allowing recourse to a certain extent. Then, instead of \( z, p \) and \( Q \) become variables in the formulation. As we will show, it is still a restriction; however, this is a major improvement compared to the case without any recourse, and as it turns out, this linear approximation of recourse can often achieve the same level of flexibility.

Substituting the affine function of \( \theta \) for \( z(\theta) \), the restricted flexibility test problem can be formulated as:

\[
\bar{\chi}(d) = \min_{p \in \mathbb{R}^{nz}, Q \in \mathbb{R}^{nz \times n\theta}} \max_{\theta \in T} \max_{j \in J} f_j(d, p, Q, \theta) = \min_{p \in \mathbb{R}^{nz}, Q \in \mathbb{R}^{nz \times n\theta}} \max_{\theta \in T} \min_{u \in \mathbb{R}} u
\]

\[
\text{s.t. } Ad + B(p + Q\theta) + C\theta \leq u e, \quad \tag{8.9a}
\]

Proposition 8.2. Problem (8.9) has its optimal solution at a vertex of \( T \).

Proof. Problem (8.9) can be equivalently written as:

\[
\bar{\chi}(d) = \min_{p \in \mathbb{R}^{nz}, Q \in \mathbb{R}^{nz \times n\theta}} \max_{\theta \in T} \min_{z, u} u \quad \tag{8.10a}
\]

\[
\text{s.t. } Ad + Bz + C\theta \leq u e \quad \tag{8.10b}
\]

\[
z - (p + Q\theta) \leq 0 \quad \tag{8.10c}
\]

\[
-z + (p + Q\theta) \leq 0 \quad \tag{8.10d}
\]

\[
z \in \mathbb{R}^{nz}, u \in \mathbb{R}, \quad \tag{8.10e}
\]

where the affine control function has been incorporated by adding the two inequality constraints (8.10c) and (8.10d). For any fixed \( p \) and \( Q \), the constraint functions are jointly convex in \( z \) and \( \theta \). Therefore, according to Theorem 8.1, the solution lies at a vertex of \( T \) for any \( p \) and \( Q \). In particular, this holds true for the \( p \) and \( Q \) that minimize the objective function. Hence, Problem (8.10) has its optimal solution at a vertex of \( T \), which is equivalently true for Problem (8.9).

Proposition 8.2 implies that we only need to consider the vertices of \( T \), of which
there are finitely many. This allows us to obtain \( \bar{\chi}(d) \) by solving:

\[
\bar{\chi}(d) = \min_{p, Q, u} \ u
\]
\[
\text{s.t. } A d + B (p + Q \hat{\theta}_t) + C \hat{\theta}_t \leq u \quad \forall t \in \mathcal{T} \tag{FT_{\text{AARO}}}
\]

\[
p \in \mathbb{R}^{n_z}, \quad Q \in \mathbb{R}^{n_z \times n_{\theta}}, \quad u \in \mathbb{R},
\]

where \( \mathcal{T} \) is the set of vertices of \( T \).

**Proposition 8.3.** Let \( \chi(d) \) and \( \bar{\chi}(d) \) be the flexibility function values obtained from solving Problems (FT_{TFA}) (or alternatively (FT_{DFA})) and (FT_{\text{AARO}}), respectively. Then, \( \chi(d) \leq \bar{\chi}(d) \).

**Proof.** Solving Problem (FT_{\text{AARO}}) is equivalent to solving Problem (FT_{TFA}) with the additional constraint \( z = p + Q \theta \) where \( p \in \mathbb{R}^{n_z}, \quad Q \in \mathbb{R}^{n_z \times n_{\theta}} \). Hence, (FT_{\text{AARO}}) is a restriction of (FT_{TFA}), from which it follows that \( \chi(d) \leq \bar{\chi}(d) \).

**Proposition 8.4.** For \( n_{\theta} = 1 \), \( \chi(d) = \bar{\chi}(d) \).

**Proof.** Let \( z^1 \) and \( z^2 \) be the solutions of Problem (FF) for \( \theta = \theta^L \) and \( \theta = \theta^U \), respectively. According to Theorem 8.1, \( \chi(d) \) will be obtained at one of these two vertex solutions. By formulating the affine control function

\[
z = \begin{cases} 
  z^1 & \text{if } \frac{z^2 - z^1}{\theta^U - \theta^L} \leq \frac{z^2 - z^1}{\theta^U - \theta^L} \\
  \left(\frac{z^2 - z^1}{\theta^U - \theta^L}\right) \theta^L + \left(\frac{z^2 - z^1}{\theta^U - \theta^L}\right) \theta^U
\end{cases}
\]

which corresponds to a line in the \((1 + n_z)\)-dimensional \((\theta, z)\)-space, both vertex solutions can be considered. This implies that when solving Problem (FT_{\text{AARO}}), the same controls \( z^1 \) and \( z^2 \) can be applied at the vertices of \( T \). According to Proposition 8.2, \( \bar{\chi}(d) \) will be obtained at one of these two vertex solutions. Hence, \( \chi(d) = \bar{\chi}(d) \). An illustrative example with a scalar \( z \) is shown in Figure 8.3.

Following a similar argument, one arrives at a more general statement:

**Proposition 8.5.** If the hyperbox uncertainty set \( T \) is replaced by a polytope with \( n_{\theta} + 1 \) vertices, denoted by \( \mathcal{T} \), then \( \chi(d) = \bar{\chi}(d) \).

**Proof.** Suppose \( z^1, z^2, \ldots, z^{n_{\theta}+1} \) are the solutions of Problem (FF) at the \( n_{\theta} + 1 \) vertices of \( \mathcal{T} \). All \( n_{\theta} + 1 \) vertex solutions can be expressed through an affine control function in the form of \( z = p + Q \theta \), which represents an \( n_{\theta} \)-dimensional hyperplane in the \((n_{\theta} + n_z)\)-dimensional \((\theta, z)\)-space. Thus, the same solution will be obtained by solving Problems (FT_{TFA}) and (FT_{\text{AARO}}). Hence, \( \chi(d) = \bar{\chi}(d) \).
8. Relation Between Flexibility Analysis and Robust Optimization

However, since uncertainty sets in the flexibility analysis problems are hyperboxes, Proposition 8.5 does not apply except in the case of \( n_\theta = 1 \). For \( n_\theta \geq 2 \), the number of vertices of the box uncertainty set is greater than \( n_\theta + 1 \). Then, we may encounter the situation in which not all vertex solutions of the original flexibility test problem can be expressed by an affine control function. In that case, the restricted flexibility test formulation (8.9) may fail in obtaining the same optimal solution. To formalize this intuitive result, we state the following:

Proposition 8.6. For \( n_\theta \geq 2 \), there exist constraint matrices \( A, B, C \), and designs \( d \) such that \( \chi(d) < \bar{\chi}(d) \).

Proof. We prove this proposition by finding an example for \( n_\theta = 2 \) in which \( \chi(d) < \bar{\chi}(d) \). The same result then follows for \( n_\theta > 2 \) since the case of \( n_\theta = 2 \) can be seen as a special case of \( n_\theta > 2 \).

Consider the example with the following four constraints:

\[
\begin{align}
  f_1 &= -z \leq 0 & (8.12a) \\
  f_2 &= z - \theta_2 \leq 0 & (8.12b) \\
  f_3 &= z - \theta_1 \leq 0 & (8.12c) \\
  f_4 &= -20 - z + \theta_1 + \theta_2 \leq 0 & (8.12d)
\end{align}
\]

which form the polytope that is shown in Figure 8.4. The uncertainty set is chosen to be \( T = \{(\theta_1, \theta_2) : 0 \leq \theta_1 \leq 20, \ 0 \leq \theta_2 \leq 20\} \).

By solving Problem (FT\(_{DFA}\)) for this example, we obtain \( \chi = 0 \). However, \( \bar{\chi} = 5 \) is obtained by solving Problem (FT\(_{AARO}\)). Thus, in this particular case with \( n_\theta = 2 \),
8. Relation Between Flexibility Analysis and Robust Optimization

Figure 8.4: Illustrative example with \( n_\theta = 2 \), where \( \chi(d) < \bar{\chi}(d) \).

we have \( \chi < \bar{\chi} \), which proves the proposition.

In Figure 8.4, one can see that for each of the vertices of \( T \), there is exactly one feasible \( z \). A plane in the 3-dimensional \((\theta, z)\)-space, which could be described by an affine control function, can cover at most three of the four vertex solutions. Therefore, while the process is feasible for the entire uncertainty set if \( z \) can be freely adjusted within the feasible region, this cannot be achieved if \( z \) is restricted to be an affine function of \( \theta \).

Another geometric interpretation of Proposition 8.6 can be obtained by examining the projection of the feasible region onto the \( \theta \)-space. In our example, as one can see in Figure 8.4, the projection of the 3-dimensional polytope representing the feasible region onto the 2-dimensional \( \theta \)-space covers exactly \( T \); thus, \( \chi = 0 \). However, one cannot find a plane such that the projection of the intersection between the plane and the polytope covers \( T \); hence, \( \bar{\chi} > 0 \).

Formulating Robust Counterpart

If \( T \) has a large number of vertices, Problem \((FT_{\text{ARO}})\) can quickly become computationally intractable. In order to avoid enumerating all vertices, we first notice that the restricted flexibility test problem given by Eq. (8.9a) can be equivalently
written as follows by interchanging the two inner maximizations:

\[
\bar{\chi}(d) = \min_{p \in \mathbb{R}^{n_z}, Q \in \mathbb{R}^{n_z \times n_\theta}} \max_{j \in J} \max_{\theta \in T} f_j(d, p, Q, \theta),
\]

from which it follows that we arrive at the same solution by applying a constraint-wise worst-case approach, i.e. by taking \( \max_{\theta \in T} f_j \) for each individual \( j \in J \). Hence, the restricted flexibility test problem can be reformulated as follows:

\[
\bar{\chi}(d) = \min_{p, Q, u} u
\]

s.t. \( a_j d + b_j p + \max_{\theta \in T} \{(b_j Q + c_j) \theta\} \leq u \quad \forall j \in J \)

\( p \in \mathbb{R}^{n_z}, Q \in \mathbb{R}^{n_z \times n_\theta}, u \in \mathbb{R} \).

Problem (8.14) is a bilevel problem with \( m \) lower-level problems. For each \( j \), the corresponding lower-level problem is:

\[
\max_{\theta} \quad (b_j Q + c_j) \theta
\]

s.t. \( \theta^L \leq \theta \leq \theta^U \),

of which the dual is:

\[
\min_{\mu_j, \nu_j} \quad \left( \theta^U \right)^T \mu_j - \left( \theta^L \right)^T \nu_j
\]

s.t. \( \mu_j - \nu_j = (b_j Q + c_j)^T \)

\( \mu_j \in \mathbb{R}^{n_\theta}, \nu_j \in \mathbb{R}^{n_\theta} \).

By substituting the dual formulations of the lower-level problems into Problem (8.14), we obtain the following single-level problem, which is generally referred to as the affinely adjustable robust counterpart (Ben-Tal et al., 2009):

\[
\bar{\chi}(d) = \min_{p, Q, u, \mu, \nu} u
\]

s.t. \( a_j d + b_j p + \left[ \left( \theta^U \right)^T \mu_j - \left( \theta^L \right)^T \nu_j \right] \leq u e \quad \forall j \in J \) (FT\text{AARO})

\( \mu_j - \nu_j = (b_j Q + c_j)^T \quad \forall j \in J \)

\( p \in \mathbb{R}^{n_z}, Q \in \mathbb{R}^{n_z \times n_\theta}, u \in \mathbb{R}, \mu \in \mathbb{R}^{n_\theta \times m}, \nu \in \mathbb{R}^{n_\theta \times m} \),

which is an LP with \( m(n_\theta + 1) \) constraints and \( n_\theta(2m + n_z) + n_z + 1 \) variables. Due to strong duality, Problems (8.14) and (FT\text{AARO}) have the same objective function.
value at the optimal solution.

8.4. Flexibility Index Problem

8.4.1. Problem Statement

Define the uncertainty set as $T(\delta) = \{\theta : \theta^N - \delta \Delta \theta^- \leq \theta \leq \theta^N + \delta \Delta \theta^+\}$, where $\delta$ is a nonnegative scalar, and $\Delta \theta^-$ and $\Delta \theta^+$ are incremental negative and positive deviations from the nominal value, respectively. The flexibility index problem (Swaney & Grossmann, 1985a) can then be stated as follows: For a given design $d$, find the largest $\delta$ such that by proper adjustment of the control variables $z$, the inequalities $f_j(d, z, \theta) \leq 0$, $j \in J$, hold for all $\theta \in T(\delta)$. This maximum $\delta$ is referred to as the flexibility index $F(d)$.

8.4.2. Traditional Flexibility Analysis

The flexibility index can be computed as follows (Swaney & Grossmann, 1985a):

$$F(d) = \max_{\delta \in \mathbb{R}_+} \delta$$

subject to

$$\max_{\theta \in T(\delta)} \min_{z \in \mathbb{R}^n} \max_{j \in J} f_j(d, z, \theta) \leq 0,$$

which is equivalent to the minimum $\delta$ for which the feasibility function $\psi(d, \theta)$ is zero, i.e.

$$F(d) = \min_{\delta \in \mathbb{R}_+, \theta \in T(\delta)} \delta$$

subject to $\psi(d, \theta) = 0$.

By applying the active-set method (Grossmann & Floudas, 1987), Problem (8.18) can be formulated as the following MILP:

$$F(d) = \min_{\theta, z, \lambda, s, y, \delta} \delta$$

subject to

$$Ad + Bz + C\theta + s = 0$$
$$e^T\lambda = 1$$
$$B^T\lambda = 0$$
$$\lambda \leq y$$
$$s \leq M(e - y)$$

(FI_TFA)
8. RELATION BETWEEN FLEXIBILITY ANALYSIS AND ROBUST OPTIMIZATION

\[ e^T y \leq n_z + 1 \]
\[ \theta^N - \delta \Delta \theta^- \leq \theta \leq \theta^N + \delta \Delta \theta^+ \]
\[ z \in \mathbb{R}^{n_z}, \lambda \in \mathbb{R}^m, s \in \mathbb{R}^m, y \in \{0, 1\}^m, \delta \in \mathbb{R}_+, \]

which has the same numbers of constraints and variables as Problem (FTTFA).

8.4.3. Duality-based Flexibility Analysis

Define \( \theta = \theta^N + \delta \tilde{\theta} \) with \( \tilde{\theta} \in \tilde{T} \), where
\[
\tilde{T} = \{ \tilde{\theta} : \tilde{\theta}_i = x_i \Delta \theta_i^+ - (1 - x_i) \Delta \theta_i^- , x_i \in \{0, 1\} \quad \forall i \in \Theta \} .
\] (8.20)

The flexibility index problem can then be stated as:

\[
F(d) = \min_{\tilde{\theta} \in \tilde{T}(\delta)} \max_{\delta, z, \tilde{\theta}} \delta
\]
\[ \text{s.t. } A d + B z + C (\theta^N + \delta \tilde{\theta}) \leq 0 \] (8.21a)
\[ \begin{align*}
& z \in \mathbb{R}^{n_z}, \delta \in \mathbb{R}_+, \\
& \tilde{\theta} \in \tilde{T}, \lambda \in \mathbb{R}_+.
\end{align*} \] (8.21b)

where the dual of the inner maximization problem for a given \( \tilde{\theta} \) is:

\[
\delta(d, \tilde{\theta}) = \min_{\lambda} \quad \left( -A d - C \theta^N \right)^T \lambda
\]
\[ \text{s.t. } B^T \lambda = 0 \] (8.22a)
\[ \tilde{\theta}^T C^T \lambda \geq 1 \] (8.22b)
\[ \lambda \in \mathbb{R}_+. \] (8.22c)

\[
\text{The flexibility index problem then becomes:}
\]
\[
F(d) = \min_{\delta, \lambda} \quad \left( -A d - C \theta^N \right)^T \lambda
\]
\[ \text{s.t. } B^T \lambda = 0 \] (8.23a)
\[ \tilde{\theta}^T C^T \lambda \geq 1 \] (8.23b)
\[ \tilde{\theta} \in \tilde{T}, \lambda \in \mathbb{R}_+. \] (8.23c)

242
which following a similar reasoning as in (FT_{DFA}) can be reformulated into:

\[
F(d) = \min_{\lambda, \bar{\lambda}, x} \left(-A d - C \theta^N\right)^T \lambda
\]

s.t. \(B^T \lambda = 0\)

\[
\sum_{j \in J} \sum_{i \in \Theta} c_{ji} \left[-\lambda_j \Delta \theta_i^- + \bar{\lambda}_{ij} \left(\Delta \theta_i^+ + \Delta \theta_i^-\right)\right] \geq 1
\]

\[
\bar{\lambda}_{ij} \geq (\lambda_j - 1) + x_i \quad \forall \ i \in \Theta, \ j \in J
\]

\[
\bar{\lambda}_{ij} \leq \lambda_j \quad \forall \ i \in \Theta, \ j \in J
\]

\[
\bar{\lambda}_{ij} \leq x_i \quad \forall \ i \in \Theta, \ j \in J
\]

\[
\lambda \in \mathbb{R}^m, \ \bar{\lambda} \in \mathbb{R}^{n_\theta \times m}, \ x \in \{0, 1\}^{n_\theta},
\]

which has the same numbers of constraints and variables as Problem (FT_{DFA}).

### 8.4.4. Affinely Adjustable Robust Optimization

If we follow a similar approach as in Section 8.3.4 to derive the AARO formulation, but directly apply the uncertainty set \(T(\delta)\), we will arrive at a nonlinear formulation due to the dependence of the uncertainty set on \(\delta\), which is a decision variable. In order to avoid this nonlinearity, we define

\[
\theta = \theta^N - \delta \Delta \theta^- + w \delta(\Delta \theta^- + \Delta \theta^+)
\]

(8.24)

where \(w\) is the vector of normalized uncertain parameters for which the uncertainty set can be written as

\[
W = \{ w : 0 \leq w \leq e, w \in \mathbb{R}^{n_\theta} \}.
\]

(8.25)

Note that \(W\) is a fixed uncertainty set. We now choose the affine control function to be \(z = p + Q w\), so the restricted flexibility index problem becomes:

\[
\overline{F}(d) = \max_{p, Q, \delta} \delta
\]

s.t. \(a_j d + b_j p + c_j (\theta^N - \delta \Delta \theta^-)\)

\[
+ \max_{w \in W} \left\{ [b_j Q + c_j \delta(\Delta \theta^- + \Delta \theta^+)] w \right\} \leq 0 \quad \forall \ j \in J
\]

(8.26a)

\[
p \in \mathbb{R}^{n_z}, \ Q \in \mathbb{R}^{n_z \times n_\theta}, \ \delta \in \mathbb{R}_+.
\]

(8.26c)
The dual of the maximization problem in each constraint $j$ is:

$$\min_{\mu_j} \ e^T \mu_j$$ (8.27a)

s.t. $$\mu_j \geq [b_j Q + c_j \delta(\Delta \theta^- + \Delta \theta^+)]^T$$ (8.27b)

$$\mu_j \in \mathbb{R}^{n_\theta}_+,$$ (8.27c)

which leads to the following affinely adjustable robust counterpart of the flexibility index problem:

$$\overline{F}(d) = \max_{p,Q,\mu,\delta} \delta$$ (FI\textsubscript{AARO})

s.t. $$a_j d + b_j p + c_j (\theta^N - \delta \Delta \theta^-) + e^T \mu_j \leq 0 \ \forall \ j \in J$$

$$\mu_j \geq [b_j Q + c_j \delta(\Delta \theta^- + \Delta \theta^+)]^T \ \forall \ j \in J$$

$$p \in \mathbb{R}^{n_z}, Q \in \mathbb{R}^{n_z \times n_\theta}, \mu \in \mathbb{R}^{n_\theta \times m}, \delta \in \mathbb{R}_+,$$

which is an LP with $m(n_\theta + 1)$ constraints and $n_\theta(n + m) + n_z + 1$ variables.

8.5. Design Under Uncertainty with Flexibility Constraints

8.5.1. Problem Statement

The design under uncertainty problem with flexibility constraints (Halemane & Grossmann, 1983) is formulated as follows:

$$\eta = \min_{\hat{c},\bar{c}} \hat{c}^T d + \sum_{s \in S} \varphi_s \bar{c}^T \bar{z}_s$$ (8.28a)

s.t. $$A d + B \bar{z}_s + C \bar{\theta}_s \leq 0 \ \forall \ s \in S$$ (8.28b)

$$\max_{\theta \in T} \min_{z \in \mathbb{R}^{n_z}} \max_{j \in J} \{a_j d + b_j z + c_j \theta\} \leq 0$$ (8.28c)

$$d \in \mathbb{R}^n, \bar{z} \in \mathbb{R}^{n_z \times h},$$ (8.28d)

where $\hat{c}$ and $\bar{c}$ are vectors of cost coefficients. The objective function is an approximation of the total expected cost, which consists of two parts: the capital cost associated with the design, $\hat{c}^T d$, and a scenario-based approximation of the expected operating cost, $\sum_{s \in S} \varphi_s \bar{c}^T \bar{z}_s$. In the scenario set $S = \{1, 2, \ldots, h\}$, each scenario is denoted by the index $s$ and is characterized by $\bar{\theta}_s$, the value that the uncertain parameter $\theta$ takes in this particular scenario, and the corresponding probability $\varphi_s$, for which $\sum_{s \in S} \varphi_s = 1$. For each scenario, a different control, $\bar{z}_s$, is determined,
which satisfies Eq. (8.28b). The expected operating cost is approximated by calculating the sum of the operating costs for the \( h \) representative scenarios in \( S \), each weighted with the corresponding probability.

In addition to Eq. (8.28b), which ensures that the design is feasible for all pre-elected scenarios, the flexibility constraints given by Eq. (8.28c) further guarantee feasibility for all \( \theta \in T \) given that the control variable \( z \) can be adjusted depending on the realization of the uncertain parameter.

8.5.2. Flexibility Analysis

Halemane & Grossmann (1983) solve Problem (8.28) with an iterative column-and-constraint generation approach. The algorithm relies on the fact that a design is feasible for all \( \theta \in T \) if it is feasible for the worst-case realization of the uncertainty, which lies at one of the vertices of \( T \). In each iteration, a relaxation of Problem (8.28) is solved, where Eq. (8.28c) is replaced by a set of constraints of the form \( f_j(d, \hat{z}_t, \hat{\theta}_t) \leq 0 \forall j \in J \), where \( \hat{\theta}_t \) is a vertex of \( T \). Then, a flexibility test is performed for the obtained design. If the design is feasible, the algorithm terminates; otherwise, the critical point (another vertex) obtained from the flexibility test is added to the formulation, which is solved in the next iteration to determine the next design suggestion. The complete algorithm is as follows:

**Step 1** Set \( k = 0 \). Choose an initial set \( \hat{T}_0 \) consisting of \( N_0 \) critical points.

**Step 2** Solve the following problem:

\[
\min_{d, \bar{z}, \hat{z}} \quad c^T d + \sum_{s \in S} \varphi_s c^T \bar{z}_s \tag{8.29a}
\]

s.t. \( A d + B \bar{z}_s + C \hat{\theta}_s \leq 0 \quad \forall s \in S \tag{8.29b} \)

\( A d + B \hat{z}_t + C \hat{\theta}_t \leq 0 \quad \forall t \in \hat{T}_k \tag{8.29c} \)

\( d \in \mathbb{R}^n, \bar{z} \in \mathbb{R}^{n \times h}, \hat{z} \in \mathbb{R}^{n \times N_k} \tag{8.29d} \)

To obtain design \( d_k \).

**Step 3** Solve Problem \((FT_{\text{TFA}})\) or \((FT_{\text{DFA}})\) by setting \( d = d_k \), and obtain critical point \( \theta^c_k \). If \( \chi(d_k) \leq 0 \), stop; otherwise, go to Step 4.

**Step 4** Set \( \hat{\theta}_{k+1} = \theta^c_k, N_{k+1} = N_k + 1 \), and define \( \hat{T}_{k+1} = \{1, 2, \ldots, N_{k+1}\} \). Set \( k = k + 1 \) and go to Step 2.
The algorithm converges in a finite number of iterations since there is a finite number of vertices. Note that the only difference between the traditional and the duality-based flexibility analysis approaches is the flexibility test problem that is solved in Step 3.

There are different approaches for choosing the initial set of critical points \( \hat{T}_0 \). For example, one effective strategy (Grossmann & Sargent, 1978) is to determine the critical points for each constraint by examining the signs of the coefficients in matrix \( C \) and use these points as the initial set. For the remainder of this chapter, we refer to the variant of the algorithm applying this initialization strategy as DF\(_{TFA}^*\) or DF\(_{DFA}^*\) (depending on whether the TFA or DFA flexibility test problem is solved). Alternatively, one could also simply let \( \hat{T}_0 \) be empty; this variant of the algorithm will be denoted by DF\(_{TFA}^0\) or DF\(_{DFA}^0\).

It is worth mentioning that very recently, similar column-and-constraint generation algorithms have been proposed to solve the two-stage robust optimization problem that allows full adjustability in the recourse (Zhao & Zeng, 2012; Zeng & Zhao, 2013). This is just another indicator for the strong connection between flexibility analysis and robust optimization, and example of the pioneering work in flexibility analysis that has long preceded the era of robust optimization.

### 8.5.3. Affinely Adjustable Robust Optimization

Unlike the flexibility analysis approach, the AARO approach does not require an iterative framework. Here, we only need to solve a single LP. In order to obtain the AARO formulation, we simply take Problem (FT\(_{AARO}\)), set \( u = 0 \) to ensure feasibility, and add Eqs. (8.28a) and (8.28b) to describe the objective function. Hence, we arrive at the following formulation:

\[
\eta = \min_{d, \bar{z}, p, Q, \mu, \nu} \quad \bar{c}^T d + \sum_{s \in S} \varphi_s \bar{c}^T \bar{z}_s \\
\text{s.t.} \quad A d + B \bar{z}_s + C \bar{\theta}_s \leq 0 \quad \forall \ s \in S \\
\quad a_j d + b_j p + \left[ (\theta^U) \mu_j - (\theta^L) ^T \nu_j \right] \leq 0 \quad \forall \ j \in J \\
\quad \mu_j - \nu_j = (b_j Q + c_j) ^T \quad \forall \ j \in J \\
\quad d \in \mathbb{R}^{n_d}, \ \bar{z} \in \mathbb{R}^{n_z \times h}, \ p \in \mathbb{R}^{n_z}, \ Q \in \mathbb{R}^{n_z \times n_q}, \ \mu \in \mathbb{R}^{n_q \times m}, \ \nu \in \mathbb{R}^{n_q \times m}.
\]

Note that in Problem (DF\(_{AARO}\)), the control variables for the scenarios used to
compute the expected cost and the ones used to guarantee feasibility for all $\theta \in T$ are treated differently. While $\tilde{z}_s$ are fully adjustable, the adjustable controls in the flexibility constraints are restricted to affine functions.

For the sake of comparison, we also present a formulation in which $\tilde{z}_s$ are set according to the same affine functions:

$$
\eta' = \min_{d,p,Q,\mu,\nu} \quad c^T d + \sum_{s \in S} \varphi_s c^T \left( p + Q \bar{\theta}_s \right) \\
\text{s.t.} \quad A d + B \left( p + Q \bar{\theta}_s \right) + C \bar{\theta}_s \leq 0 \quad \forall s \in S \\
a_j d + b_j p + \left[ \left( \theta^U \right)^T \mu_j - \left( \theta^L \right)^T \nu_j \right] \leq 0 \quad \forall j \in J \\
\mu_j - \nu_j = \left( b_j Q + c_j \right)^T \forall j \in J \\
d \in \mathbb{R}^{n_d}, p \in \mathbb{R}^{n_z}, Q \in \mathbb{R}^{n_z \times n_q}, \mu \in \mathbb{R}_+^{n_q \times m}, \nu \in \mathbb{R}_+^{n_q \times m}.
$$

Let $\eta$ and $\eta'$ be the total expected costs at the optimal solutions of Problems (DF$_{AARO}$) and (DF'$_{AARO}$), respectively. Clearly, since Problem (DF'$_{AARO}$) is a restriction of Problem (DF$_{AARO}$), $\eta \leq \eta'$. The advantage of Problem (DF'$_{AARO}$) is that it does not involve the variables $\tilde{z}_s$, which, however, usually does not translate into significant reductions in computation times.

### 8.6. Numerical Results

In the following, we apply the proposed models to flexibility analysis problems for three different examples. While the first two examples are meant to be illustrative, the third one is significantly larger in size and allows more insights into the computational performance of the different approaches. All models were implemented in GAMS 24.4.1 (GAMS Development Corporation, 2015a), and the commercial solver CPLEX 12.6.1 (IBM ILOG, 2015a) was applied to solve the LPs and MILPs on an Intel® Core™ i7-2600 machine at 3.40 GHz with 8 processors and 8 GB RAM running Windows 7 Professional. Unless specified otherwise, the LPs were solved with the concurrent option using all 8 threads. Similarly, MILPs were also solved in parallel using all available threads.

#### 8.6.1. Example 1: Heat Exchanger Network

This first example of a small heat exchanger network (HEN) is based on an example introduced by Grossmann & Floudas (1987). The HEN consisting of two hot streams, two cold streams, three heat exchangers and one cooler is shown in
Figure 8.5. Here, the uncertainty lies in the inlet temperatures of the hot and cold streams; hence, the four uncertain parameters are $T_1$, $T_3$, $T_5$, and $T_8$. The nominal values for the uncertain temperatures are shown in Figure 8.5, and the maximum deviations from the nominal values for each temperature are assumed to be ±5 K. The control variable is the heat load in the cooler denoted by $Q_C$. We consider as design variable $T_7^{\text{max}}$, which is the upper bound on $T_7$.

The five inequality constraints for the given HEN are the following:

\begin{align*}
-350 - 0.67Q_C + T_3 &\leq 0 & \quad \text{(8.30a)} \\
1388.5 + 0.5Q_C - 0.75T_1 - T_3 - T_5 &\leq 0 & \quad \text{(8.30b)} \\
2044 + Q_C - 1.5T_1 - 2T_3 - T_5 &\leq 0 & \quad \text{(8.30c)} \\
2830 + Q_C - 1.5T_1 - 2T_3 - T_5 - 2T_8 &\leq 0 & \quad \text{(8.30d)} \\
-2830 - T_7^{\text{max}} - Q_C + 1.5T_1 + 2T_3 + T_5 + T_8 &\leq 0 & \quad \text{(8.30e)}
\end{align*}

For the initial design with $T_7^{\text{max}} = 317$ K, we solve the flexibility test and flexibility index problems by applying the proposed models from the TFA, DFA, and AARO approaches. The results are shown in Tables 8.1 and 8.2. Notice that all three approaches obtain the same results, which implies that in this case, the same
level of flexibility can be achieved by restricting the control variable to take the form of an affine function of the uncertain parameters. Tables 8.1 and 8.2 also show the numbers of constraints, continuous, and binary variables in each model, which indicate the small size of this problem. As a result, the solution times are marginal.

**Table 8.1: Flexibility test results for Example 1.**

<table>
<thead>
<tr>
<th>Model</th>
<th>$\chi(d)$, $\bar{\chi}(d)$</th>
<th># of Constraints</th>
<th># of Cont. Variables</th>
<th># of Bin. Variables</th>
<th>Solution Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>($F_{TFA}$)</td>
<td>2.0</td>
<td>26</td>
<td>16</td>
<td>5</td>
<td>0.11</td>
</tr>
<tr>
<td>($F_{DFA}$)</td>
<td>2.0</td>
<td>62</td>
<td>25</td>
<td>4</td>
<td>0.06</td>
</tr>
<tr>
<td>($F_{AARO}$)</td>
<td>2.0</td>
<td>25</td>
<td>46</td>
<td></td>
<td>0.05</td>
</tr>
</tbody>
</table>

**Table 8.2: Flexibility index results for Example 1.**

<table>
<thead>
<tr>
<th>Model</th>
<th>$F(d)$, $\bar{F}(d)$</th>
<th># of Constraints</th>
<th># of Cont. Variables</th>
<th># of Bin. Variables</th>
<th>Solution Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>($F_{TFA}$)</td>
<td>0.7</td>
<td>26</td>
<td>16</td>
<td>5</td>
<td>0.06</td>
</tr>
<tr>
<td>($F_{DFA}$)</td>
<td>0.7</td>
<td>62</td>
<td>25</td>
<td>4</td>
<td>0.05</td>
</tr>
<tr>
<td>($F_{AARO}$)</td>
<td>0.7</td>
<td>25</td>
<td>26</td>
<td></td>
<td>0.03</td>
</tr>
</tbody>
</table>

Now we optimize the design while satisfying the flexibility constraints. The cost coefficients for the design variable $T_{max}^7$ and the control variable $Q_C$ are set to 2 and 1, respectively. For the approximation of the expected operating cost, three scenarios are considered for which the data are shown in Table 8.3. Note that Scenario 2 is the nominal case.

**Table 8.3: Data for discrete scenarios considered in Example 1.**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>$T_1$</th>
<th>$T_3$</th>
<th>$T_5$</th>
<th>$T_8$</th>
<th>$\varphi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>615</td>
<td>383</td>
<td>578</td>
<td>308</td>
<td>0.25</td>
</tr>
<tr>
<td>2</td>
<td>620</td>
<td>388</td>
<td>583</td>
<td>313</td>
<td>0.50</td>
</tr>
<tr>
<td>3</td>
<td>625</td>
<td>393</td>
<td>588</td>
<td>318</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Table 8.4 shows the results for the design under uncertainty problem. Recall that in the case of the iterative algorithms, superscript 0 denotes the variants in which the initial set of critical points $T_0$ is empty, while superscript * denotes the initialization strategy in which a critical point is included in $T_0$ for each constraint. The numbers of constraints and variables refer to the LP that is solved to obtain the optimal design; in the TFA and DFA approaches, this is the LP solved in the final
iteration of the algorithm. Here, one observes the trade-off between the number of iterations and the sizes of the LPs solved at each iteration. In $DF^0_{TFA}$ and $DF^0_{DFA}$, only one critical point needs to be considered, resulting in two small LPs solved in two iterations. $DF^*_{TFA}$ and $DF^*_{DFA}$ create a larger initial set of critical points, which leads to a larger LP; however, now only one iteration is required. Since a flexibility test has to be performed at each iteration in addition to solving the LP, reducing in the number of iterations can be of great computational benefit. In this particular case, the smaller number of iterations leads to shorter solution times.

<table>
<thead>
<tr>
<th>Model/Algorithm</th>
<th>Expected Cost</th>
<th># of Iterations</th>
<th># of Constraints</th>
<th># of Cont. Variables</th>
<th>Solution Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$DF^0_{TFA}$</td>
<td>724</td>
<td>2</td>
<td>20</td>
<td>5</td>
<td>0.47</td>
</tr>
<tr>
<td>$DF^*_{TFA}$</td>
<td>724</td>
<td>1</td>
<td>30</td>
<td>7</td>
<td>0.23</td>
</tr>
<tr>
<td>$DF^0_{DFA}$</td>
<td>724</td>
<td>2</td>
<td>20</td>
<td>5</td>
<td>0.49</td>
</tr>
<tr>
<td>$DF^*_{DFA}$</td>
<td>724</td>
<td>1</td>
<td>30</td>
<td>7</td>
<td>0.24</td>
</tr>
<tr>
<td>($DF^0_{AARO}$)</td>
<td>724</td>
<td>40</td>
<td>49</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>($DF^*_{AARO}$)</td>
<td>727</td>
<td>40</td>
<td>46</td>
<td>0.09</td>
<td></td>
</tr>
</tbody>
</table>

By solving Problem ($DF^0_{AARO}$), we immediately arrive at the optimal solution. No iterations are required and no additional flexibility test needs to be performed; hence, although the LP to be solved is larger, the solution time is significantly less than the ones required by the TFA and DFA algorithms. Furthermore, one can see the difference between Problems ($DF^0_{AARO}$) and ($DF^*_{AARO}$), where the latter leads to a slightly higher total expected cost due to the restriction of the control variable to an affine function also in the three scenarios used to compute the expected cost.

### 8.6.2. Example 2: Process Flowsheet

Consider the simple process flowsheet shown in Figure 8.6. In this process, Material A is fed into a reactor where it reacts to Materials B and C at a fixed conversion ratio $\xi$. Materials B and C are then separated in an ideal separator in order to satisfy product demands $D_B$ and $D_C$. Here, the uncertainty lies in the demands and in the effective geometries of the inlet pipe and of the reactor, which have an impact on the flow capacities in the pipe and the reactor. The geometric properties of the pipe and the reactor are represented by the characteristic numbers $R$ and $V$, respectively; these parameters may be uncertain due to fouling or wear and tear. The control variable here is the feed flowrate $m_A$. 
8. Relation Between Flexibility Analysis and Robust Optimization

Figure 8.6: Example 2, simple process with uncertainty in product demands and equipment geometries.

The given process is represented by the following inequality constraints:

\begin{align*}
- m_A + 0.2 V &\leq 0 \quad (8.31a) \\
m_A - V &\leq 0 \quad (8.31b) \\
m_A - R &\leq 0 \quad (8.31c) \\
- 30 - m_A + 0.8 V + 1.2 R &\leq 0 \quad (8.31d) \\
- \xi m_A + D_B &\leq 0 \quad (8.31e) \\
- (1 - \xi)m_A + D_C &\leq 0 \quad (8.31f)
\end{align*}

where Eq. (8.31a) states that the reactor requires a minimum flowrate depending on its size, which at the same time imposes an upper bound on the flowrate as expressed in Eq. (8.31b). Similarly, the flowrate is limited by the size of the pipe, which is stated in Eq. (8.31c). The critical constraint is Eq. (8.31d). Here, we assume that some flow conditions, such as turbulent flow, have to be satisfied in the reactor in order for the reaction to be effective. The required flowrate depends on both the pipe and the reactor geometries; the correlation for this relationship is approximated by the linear function in Eq. (8.31d). Finally, Eqs. (8.31e) and (8.31f) state that the product flowrates have to be greater than or equal to the demands.

We solve the flexibility test and flexibility index problems for \( \xi = 0.6 \) and the following uncertainty sets: \( D_B \in [6, 8] \), \( D_C \in [3, 5] \), \( R \in [15, 25] \), and \( V \in [14, 26] \). In the nominal case, each uncertain parameter takes the value of the midpoint in the corresponding uncertainty range. The results are shown in Table 8.5. One can see that the flexibility analysis and AARO approaches obtain different solutions,
namely $\chi(d) < \bar{\chi}(d)$ and $F(d) > \bar{F}(d)$, which is due to the restriction of the control variable to affine functions in the AARO models. In this case, the TFA and DFA models correctly report that the process is feasible for every possible realization of the uncertainty given that the feed flowrate can be properly adjusted, while the AARO model fails to do so. Note that we do not report computational results due to the small problem sizes and marginal solution times as is the case in Example 1.

<table>
<thead>
<tr>
<th>Approach</th>
<th>$\chi(d), \bar{\chi}(d)$</th>
<th>$F(d), \bar{F}(d)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFA/DFA</td>
<td>-0.25</td>
<td>1.09</td>
</tr>
<tr>
<td>AARO</td>
<td>1.10</td>
<td>0.78</td>
</tr>
</tbody>
</table>

8.6.3. Example 3: Planning of a Large-Scale Process Network

In this example, we consider the long-term production planning of a large-scale process network representing a petrochemical complex (Sahinidis et al., 1989). The given process network, which is schematically shown in Figure 8.7, consists of 38 processes and 28 chemicals. The planning model is shown in the following. Note that the nomenclature for this model is independent from and therefore not to be confused with the one used for the models presented in the previous sections.

**Figure 8.7:** Schematic of the petrochemical complex considered in Example 3 (figure adapted from Park et al. (2006)). For the detailed process network, see the original paper by Sahinidis et al. (1989).
The model formulation consists of the following constraints:

\[
Q_j^0 + \sum_{t'=1}^{t} \left( \sum_{i \in \hat{I}_j} \mu_{ij} P_{it'} - \sum_{i \in \bar{I}_j} \mu_{ij} P_{it'} + W_{jt'} - D_{jt'} \right) \geq Q_j^{\text{min}} \quad \forall \ j \in J, \ t \in T \tag{8.32a}
\]

\[
Q_j^0 + \sum_{t'=1}^{t} \left( \sum_{i \in \hat{I}_j} \mu_{ij} P_{it'} - \sum_{i \in \bar{I}_j} \mu_{ij} P_{it'} + W_{jt'} - D_{jt'} \right) \leq Q_j^{\text{max}} \quad \forall \ j \in J, \ t \in T \tag{8.32b}
\]

\[
P_{it} \leq P_{i}^{\text{max}} \quad \forall \ i \in I, \ t \in T \tag{8.32c}
\]

\[
W_{jt} \leq W_{jt}^{\text{max}} - \sum_{j' \in \bar{J}_j} \xi_{jj'} D_{jt'} \quad \forall \ j \in J, \ t \in T \tag{8.32d}
\]

\[
P_{it} \geq 0 \quad \forall \ i \in I, \ t \in T \tag{8.32e}
\]

\[
W_{jt} \geq 0 \quad \forall \ j \in J, \ t \in T \tag{8.32f}
\]

where \( J, I, \) and \( T \) are the sets of chemicals, processes, and time periods, respectively. \( Q_j^0 \) is the initial inventory of chemical \( j \), and \( Q_j^{\text{min}} \) and \( Q_j^{\text{max}} \) are the minimum and maximum inventory levels, respectively. \( P_{it} \) denotes the amount of main product produced by process \( i \) in time period \( t \). The production or consumption of chemical \( j \) by process \( i \) is given by a conversion factor, denoted by \( \mu_{ij} \), with respect to the main product; hence, \( \mu_{ij} P_{it} \) is the amount of chemical \( j \) that is produced or consumed by process \( i \) in time period \( t \). The sets of processes producing and consuming chemical \( j \) are denoted by \( \hat{I}_j \) and \( \bar{I}_j \), respectively. \( W_{jt} \) denotes the amount of chemical \( j \) purchased, and \( D_{jt} \) is the demand for chemical \( j \) in time period \( t \).

Eqs. (8.32a) and (8.32b) are the inventory constraints, which ensure that the inventory level is within the given bounds at any time. Eq. (8.32c) sets the production capacity for each process. In Eq. (8.32d), the assumption is that the purchase limit for chemical \( j \) depends on the demand for products that require chemical \( j \) as feed, which is represented by the set of products \( \bar{J}_j \). The intuition is that the higher the market demand is for products requiring feed \( j \), the more limited will the availability of chemical \( j \) be. Here, the coefficient \( \xi_{jj'} \) defines how much the purchase limit for chemical \( j \) is affected by the demand of chemical \( j' \in \bar{J}_j \). Eqs. (8.32e) and (8.32f) are nonnegativity constraints.

In this model, \( Q_j^0, Q_j^{\text{min}}, W_{jt}^{\text{max}}, \mu_{ij}, \) and \( \xi_{jj'} \) are fixed constants, \( Q_j^{\text{max}} \) and \( P_i^{\text{max}} \) are the design variables, and \( P_{it} \) and \( W_{jt} \) are the control variables. The uncertainty lies in the demand, i.e. the uncertain parameters are \( D_{jt} \) of which there exist 16 for each time period.

In the following, we apply the proposed models to problem instances of differ-
ent sizes, which are created by varying the number of time periods, $N_T$. The limit on the computation time for each problem is set to one hour (wall-clock time). The flexibility test and flexibility index problems are solved for a particular design, i.e. for specific fixed values of $Q_j^{\text{max}}$ and $P_i^{\text{max}}$, the results are shown in Tables 8.6 and 8.7. In order to provide an indicator for the tightness of each MILP formulation, we show both $v^{\text{MIP}}$ and $v^{\text{RMIP}}$, which denote the objective function values at the optimal solutions of the MILP and its LP relaxation, respectively. Note that the interpretation of $v^{\text{MIP}}$ varies between the different problems; while it may refer to $\chi(d)$ or $\bar{\chi}(d)$ in the flexibility test problems or to $F(d)$ or $\overline{F}(d)$ in the flexibility index problems, it may also be none of those if the MILP is not solved to optimality. The solutions of the AARO LP models are also listed in the $v^{\text{MIP}}$-column. Furthermore, the relative optimality gap (as defined in CPLEX) is shown for each MILP. From the computational results, we make the following observations:

- In almost all instances, the DFA and the AARO models obtained the same optimal solution. The only exception is the flexibility index problem for $N_T = 10$, where Problem (FI$_{\text{DFA}}$) was not solved to optimality. The TFA approach only solved the flexibility test problem for $N_T = 1$ to optimality within the given time limit; in all other instances, the TFA models failed to find the optimal solution or in some cases even a feasible solution (see flexibility test for $N_T = 8$ and $N_T = 10$).

- For the larger instances, the time required by the AARO LP models to solve the flexibility test and flexibility index problems was often about one order of magnitude shorter than the time required by the DFA models.

- Compared to the DFA and AARO models, the TFA models exhibit significantly smaller numbers of constraints and continuous variables. However, recall that the TFA MILP models have $m$ binary variables, while the DFA MILP models have $n_\theta$ binary variables; since there are considerably more constraints than uncertain parameters in this problem, the TFA models have larger numbers of binary variables.

- The comparison between the $v^{\text{RMIP}}$-values of the TFA and DFA models indicates that the TFA models have significantly weaker relaxations, in part because of the big-M constraints. This is the primary reason for the better computational performance of the tighter DFA MILP models.

- It should be mentioned that although in many instances, $v^{\text{RMIP}} = v^{\text{MIP}}$ for the
8. Relation Between Flexibility Analysis and Robust Optimization

DFA models, it is often the case that the LP relaxation does not result in an integer solution. In such a case, the problem does not solve at the root node and further branching is needed.

- In all instances, when solving the flexibility index problem using formulation (FI\text{\_TFA}), the lower bound did not improve (i.e. remained zero) during the branch-and-bound process, even when a depth-first branching strategy was applied. This observation implies that there is sufficient flexibility in the model such that even with only one binary variable being relaxed, there exists a feasible solution with $\delta = 0$. This special structure of the problem has the consequence that a very large number of nodes in the branch-and-bound tree have to be examined in order to prove optimality.

<table>
<thead>
<tr>
<th>$N_T$</th>
<th>Model</th>
<th># of Constraints</th>
<th># of Cont. Variables</th>
<th># of Bin. Variables</th>
<th>$v^\text{RMIP}$</th>
<th>$v^\text{MIP}$</th>
<th>Gap [%]</th>
<th>Solution Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(FT\text{_TFA})</td>
<td>538</td>
<td>369</td>
<td>152</td>
<td>0.68</td>
<td>-8.24</td>
<td>0</td>
<td>31</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_DFA})</td>
<td>7345</td>
<td>2584</td>
<td>16</td>
<td>-8.24</td>
<td>-8.24</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_AARO})</td>
<td>2584</td>
<td>5682</td>
<td>-8.24</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(FT\text{_TFA})</td>
<td>1074</td>
<td>737</td>
<td>304</td>
<td>1.98</td>
<td>-8.24</td>
<td>58</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_DFA})</td>
<td>29,281</td>
<td>10,033</td>
<td>32</td>
<td>-7.00</td>
<td>-7.00</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_AARO})</td>
<td>10,032</td>
<td>22,625</td>
<td>-7.00</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(FT\text{_TFA})</td>
<td>2146</td>
<td>1473</td>
<td>608</td>
<td>3.78</td>
<td>-8.24</td>
<td>146</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_DFA})</td>
<td>116,929</td>
<td>39,520</td>
<td>64</td>
<td>-5.23</td>
<td>-5.23</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_AARO})</td>
<td>39,520</td>
<td>90,305</td>
<td>-5.23</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(FT\text{_TFA})</td>
<td>3218</td>
<td>2209</td>
<td>912</td>
<td>4.68</td>
<td>-8.24</td>
<td>157</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_DFA})</td>
<td>262,945</td>
<td>88,464</td>
<td>96</td>
<td>-4.33</td>
<td>-4.33</td>
<td>0</td>
<td>55</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_AARO})</td>
<td>88,464</td>
<td>203,041</td>
<td>-4.33</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(FT\text{_TFA})</td>
<td>4290</td>
<td>2945</td>
<td>1216</td>
<td>5.18</td>
<td>n/a</td>
<td>n/a</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_DFA})</td>
<td>467,329</td>
<td>156,864</td>
<td>128</td>
<td>-3.85</td>
<td>-3.85</td>
<td>0</td>
<td>155</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_AARO})</td>
<td>156,864</td>
<td>360,833</td>
<td>-3.85</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(FT\text{_TFA})</td>
<td>5362</td>
<td>3681</td>
<td>1520</td>
<td>5.38</td>
<td>n/a</td>
<td>n/a</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_DFA})</td>
<td>730,081</td>
<td>244,720</td>
<td>160</td>
<td>8.0</td>
<td>-3.64</td>
<td>0</td>
<td>468</td>
</tr>
<tr>
<td></td>
<td>(FT\text{_AARO})</td>
<td>244,720</td>
<td>563,681</td>
<td>-3.64</td>
<td>26</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We now consider a design problem in which the storage and production capacities, represented by the design variables $Q^\text{\_max}_j$ and $P^\text{\_max}_i$, can be expanded in order to minimize the total expected cost, which is approximated by nine representative scenarios. The problem is solved for two cases, one with $N_T = 4$ and the other with
8. Relation Between Flexibility Analysis and Robust Optimization

Table 8.7: Flexibility index results for Example 3.

<table>
<thead>
<tr>
<th>(N_T)</th>
<th>Model</th>
<th># of Constraints</th>
<th># of Cont. Variables</th>
<th># of Bin. Variables</th>
<th>(v_{\text{RMP}})</th>
<th>(v_{\text{MIP}})</th>
<th>Gap [%]</th>
<th>Solution Time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(\text{FI}_{\text{TFA}})</td>
<td>538</td>
<td>369</td>
<td>152</td>
<td>0.0</td>
<td>8.01</td>
<td>100</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{DFA}})</td>
<td>7345</td>
<td>2584</td>
<td>16</td>
<td>8.01</td>
<td>8.01</td>
<td>0</td>
<td>0.3</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{AARO}})</td>
<td>2584</td>
<td>3249</td>
<td></td>
<td>8.01</td>
<td>0</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>(\text{FI}_{\text{TFA}})</td>
<td>1074</td>
<td>737</td>
<td>304</td>
<td>0.0</td>
<td>8.01</td>
<td>100</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{DFA}})</td>
<td>29,281</td>
<td>10,033</td>
<td>32</td>
<td>4.77</td>
<td>4.77</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{AARO}})</td>
<td>10,032</td>
<td>12,898</td>
<td></td>
<td>4.77</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>(\text{FI}_{\text{TFA}})</td>
<td>2146</td>
<td>1473</td>
<td>608</td>
<td>0.0</td>
<td>11.70</td>
<td>100</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{DFA}})</td>
<td>116,929</td>
<td>39,520</td>
<td>64</td>
<td>2.45</td>
<td>3.20</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{AARO}})</td>
<td>39,520</td>
<td>51,394</td>
<td></td>
<td>3.20</td>
<td></td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>(\text{FI}_{\text{TFA}})</td>
<td>3218</td>
<td>2209</td>
<td>912</td>
<td>0.0</td>
<td>15.00</td>
<td>100</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{DFA}})</td>
<td>262,945</td>
<td>88,464</td>
<td>96</td>
<td>1.62</td>
<td>2.65</td>
<td>0</td>
<td>104</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{AARO}})</td>
<td>88,464</td>
<td>115,489</td>
<td></td>
<td>2.65</td>
<td></td>
<td>49</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(\text{FI}_{\text{TFA}})</td>
<td>4290</td>
<td>2945</td>
<td>1216</td>
<td>0.0</td>
<td>12.35</td>
<td>100</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{DFA}})</td>
<td>467,329</td>
<td>156,864</td>
<td>128</td>
<td>1.19</td>
<td>2.39</td>
<td>0</td>
<td>723</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{AARO}})</td>
<td>156,864</td>
<td>205,185</td>
<td></td>
<td>2.39</td>
<td></td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>(\text{FI}_{\text{TFA}})</td>
<td>5362</td>
<td>3681</td>
<td>1520</td>
<td>0.0</td>
<td>15.00</td>
<td>100</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{DFA}})</td>
<td>730,081</td>
<td>244,720</td>
<td>160</td>
<td>0.93</td>
<td>2.32</td>
<td>33</td>
<td>3600</td>
</tr>
<tr>
<td></td>
<td>(\text{FI}_{\text{AARO}})</td>
<td>244,720</td>
<td>320,481</td>
<td></td>
<td>2.30</td>
<td></td>
<td>408</td>
<td></td>
</tr>
</tbody>
</table>

\(N_T = 8\); the results are shown in Table 8.8. Since the formulation \((FT_{\text{TFA}})\) seems to be unsuitable for performing the flexibility test for this problem according to the results shown above, the TFA algorithm was not applied to solve the design problem. Hence, we only compare the DFA and AARO approaches.

In both instances, one can see that Algorithm \(DF^*_{\text{DFA}}\) (with initialization of critical point set) solves the problem more quickly than Algorithm \(DF^0_{\text{DFA}}\) (with empty critical point set) due to the smaller number of iterations. In Algorithm \(DF^*_{\text{DFA}}\), the design problem formulation is significantly larger because of the larger initial set of critical points; however, since it is only an LP, the added computational effort to solve it is outweighed by the benefit of solving a smaller number of flexibility test problems, which are MILPs.

In this case, solving Problem \((DF_{\text{AARO}})\) results in the same solutions as obtained by the DFA algorithms, yet at a considerably lower computational expense since only one single LP has to be solved. This difference in computational performance is especially apparent in the larger instance. By solving Problem \((DF'_{\text{AARO}})\), higher costs are achieved, which shows the suboptimality when affine control functions...
are imposed on the scenario set. Notice that for the instance with \( N_T = 8 \), the solution time for \( (DF'_{AARO}) \) is significantly larger than for \( (DF_{AARO}) \). This stark difference is due to the fact that in this particular case, Problem \( (DF'_{AARO}) \) is poorly conditioned; hence, besides scaling, higher numeric precision had to be applied in order to solve the problem.

### 8.7. Summary

In this chapter, we have examined for linear systems the relationship between flexibility analysis and robust optimization, which are two approaches to solving optimization problems under uncertainty that originated from different research communities (PSE and OR, respectively). Although these two research areas have been developed independently from each other, they do share some fundamental conceptual ideas, such as the use of polyhedral sets to describe the uncertainty and the worst-case approach to guarantee feasibility for every possible realization of the uncertainty.

To systematically establish the link between flexibility analysis and robust optimization, and to compare these different approaches, the three classical problems from flexibility analysis have been considered: the flexibility test problem, the flexibility index problem, and design under uncertainty with flexibility constraints. For LPs with a given general structure, two new solution approaches have been proposed, where the first derives duality-based reformulations of the traditional active-set MILP formulations, and the second applies the concept of affinely adjustable robust optimization.
The concepts that form the theoretical basis for the three different approaches—traditional flexibility analysis, duality-based flexibility analysis, and AARO—have been compared. It has become clear that AARO can be seen as a special case of flexibility analysis, however with a duality-based approach to solving the problems. It can be shown that in general, AARO is more restrictive and therefore may be overly conservative. However, it turns out that for LP models, it is often the case that AARO does predict the same level of flexibility as TFA and DFA.

The three different approaches have been applied to three numerical examples, verifying some of the theoretical properties of the proposed formulations. The results further show that DFA and AARO may be computationally more efficient than TFA. The DFA models exhibit a better computational performance because of the tightness of the MILP formulations. In the case of AARO, the main advantage is that only LPs need to be solved; furthermore, no iterative procedure is required for solving the design under uncertainty problem.
9. Conclusions

In this final chapter, we provide a summary of the thesis in Section 9.1, highlight the main research contributions in Section 9.2, and then close with recommendations for future work in Section 9.3.

9.1. Summary of the Thesis

In the following, we summarize the work presented in each chapter of this thesis.

Chapter 1  When introducing the concept of demand side management (DSM), we have emphasized the two perspectives on DSM: the grid operator’s perspective and the electricity consumer’s perspective. In this work, we have been mainly concerned with the industrial electricity consumer’s perspective on DSM. In the comprehensive literature review, we have identified four major challenges in industrial DSM, namely: (1) accurate modeling of operational flexibility, (2) integration of production and energy management, (3) decision-making across multiple time scales, and (4) optimization under uncertainty. As the list of references indicates, although the benefit of industrial DSM is widely acknowledged and has gained considerable interest in recent years, little research has been done in this area. In this work, we have tried to fill some of the research gaps identified in Chapter 1.

Chapter 2  In order to consider integrated optimization problems involving complex processes, we need process models that are sufficiently accurate as well as computationally efficient. In Chapter 2, we have introduced the idea of Convex Region Surrogate (CRS) models, which can be formulated as sets of mixed-integer linear constraints that provide good approximations of nonlinearities and nonconvexities. In a CRS model, the feasible region is approximated by the union of convex regions in the form of polytopes. Furthermore, for each region, the cost function is approximated by a linear function.
We have presented a two-phase algorithm for the construction of CRS models based on given data. In Phase 1, the algorithm partitions the data points into disjoint subsets such that a linear cost correlation can be obtained within an error tolerance for all points in each subset, and that the convex hulls around the points of each subset do not overlap. In Phase 2, convex regions are constructed for each subset such that the union of the convex regions describes an accurate approximation of the feasible region. The proposed algorithm relies on a series of tailored optimization problems that are solved in an iterative framework.

An extensive computational study has been conducted in order to assess the performance of the proposed algorithm on different data sets. The computational results indicate the increase in computational effort with increasing number of data points and dimensionality. Moreover, the performance of the proposed algorithm strongly depends on the level of nonlinearity and nonconvexity implied by the given data, with computation times ranging from minutes to a few hours. However, one should keep in mind that in practice, the CRS models are generated offline, which imposes less restrictions on the computational budget.

In addition, we have applied the methodology to an industrial test case drawn from a Praxair plant. Specifically, data from a real production process have been used to generate a CRS model with a feasible region in the product space and power consumption as the cost function. The case study has further demonstrated the applicability of the algorithm as well as provided insights into the impact of given data on the performance of the algorithm. The general rule is that a sufficiently large set of useful data points is required for obtaining a good result, although to a certain extent, some tolerance parameters can be adjusted in order to mitigate the impact of poor data.

**Chapter 3** The optimal scheduling of continuous power-intensive plants requires models that accurately represent the operational flexibility in the production process and the interaction with the electricity market. For this purpose, a deterministic discrete-time scheduling model has been developed in Chapter 3. The proposed MILP formulation considers process networks where each process is represented by a CRS model. A mode-based formulation has been adopted to model operational transitions. Moreover, a block contract formulation has been proposed that allows the modeling of a large variety of power contracts including commonly occurring discount and penalty contracts.

The proposed scheduling model has been applied to an illustrative example as
well as to a real-world air separation plant. The results demonstrate the capability of the model in representing the operation of continuous process networks and modeling different power contracts. This allows the optimal scheduling of power-intensive processes involving load shifting within the feasible range of operation.

From the results, we observe the expected effect of time-sensitive electricity prices; the optimal solution suggests producing more during low- and less during high-price hours, while satisfying all operational constraints. Furthermore, the proposed MILP model has proven to be computationally very efficient. Industrial-scale problems with tens of thousands of variables (continuous and binary) and hundreds of thousands of constraints can be solved to optimality typically within a minute.

Chapter 4  Uncertainty in spot electricity price and product demand has been considered in Chapter 4. In particular, we have developed a two-stage stochastic programming model that simultaneously optimizes short-term production scheduling and electricity procurement under the aforementioned uncertainty. The here-and-now decisions are the mode of operation for the production process and the amount of electricity purchased from each power contract for each time period of the scheduling horizon, while the wait-and-see decisions are the actual production rates, the amounts of products stored, the amounts of products purchased, and the amount of electricity purchased from the spot market. Risk is taken into account by incorporating the conditional value-at-risk (CVaR) into the MILP model.

To reduce the computational effort when solving large-scale problems, a scenario reduction strategy and multicut Benders decomposition have been applied. With the proposed decomposition approach, problems with approximately 3.6 million continuous variables, 3700 binary variables, and 2.7 million constraints have been solved within two hours, while the full-space model in most cases does not return a feasible solution within the same computation time.

An illustrative example and a real-world industrial air separation case demonstrate the capability of the proposed model and solution approach. Both risk-neutral optimization (minimization of total expected cost) and risk-averse optimization (maximization of a weighted sum of total expected profit and CVaR) have been considered. The case studies show significant differences between the solutions obtained from deterministic, risk-neutral, and risk-averse optimization. Especially the electricity procurement decisions highly depend on the choice of the model. When being risk-averse, considering the CVaR leads to solutions that ef-
fectively hedge against the risk of low-profit scenarios by increasing the purchase from power contracts with certain prices. Also, we have emphasized the analysis of the value of stochastic solution, which has led to the following remarkable insight: In risk-neutral optimization, accounting for electricity price uncertainty in the stochastic model does not result in significant additional benefit. In contrast, in risk-averse optimization, modeling price uncertainty is crucial for obtaining good solutions.

Chapter 5  An electricity consumer’s flexibility for DSM can be significantly increased by installing capacity for storing electric energy. In Chapter 5, we have considered cryogenic energy storage (CES), where energy is stored in the form of liquefied gas. Although CES on an industrial scale is a relatively new approach, the technology is well-known and essentially part of any air separation unit (ASU) that utilizes cryogenic separation. Here, the objective has been the assessment of the operational benefits of adding a CES system to an existing cryogenic air separation plant. We have developed an MILP scheduling model for an integrated ASU-CES plant that incorporates the possibility of recovering energy from CES for internal use or for being sold to the electric energy market.

Using a robust optimization approach, the model has been further extended to account for uncertainty in operating reserve demand. This allows the model to consider reserve market participation and yield solutions that guarantee reserve dispatch feasibility under the committed reserve capacity. The reserve demand is an endogenous uncertain parameter since it is bounded by the amount of committed reserve capacity, which is a decision variable. To consider this endogenous uncertainty, we have applied a decision-dependent uncertainty set, which further includes budget parameters that can be used to adjust the level of conservatism in the solution.

The proposed model has been applied in a real-world industrial case study. The results exhibit typical relative cost savings of approximately 10% under relatively conservative CES efficiency and uncertainty assumptions. A sensitivity analysis shows that besides the CES efficiency, economic benefits strongly depend on the level of plant utilization. If the level of utilization is low, which allows high flexibility for load shifting, cost reduction up to over 20% (for 40% utilization) can be achieved. This suggests that an added CES system may be an especially good option for underutilized air separation plants. The computation time required to solve the robust model has proven to be short (less than a minute) and in the same
order as the time required for the corresponding deterministic model.

**Chapter 6** Operating reserve can be provided not only by power generation facilities, but also by consumers that can quickly reduce their electricity consumption. The latter type of operating reserve is also referred to as interruptible load. In Chapter 6, we have developed an adjustable robust scheduling model for continuous industrial processes providing interruptible load. Unlike in Chapter 5, where we have applied static robust optimization, which does not account for recourse, the proposed adjustable robust MILP model incorporates recourse decisions by stating them as functions of the uncertain parameters, also referred to as decision rules. Multistage decision-making is considered by constructing the decision rules such that a recourse variable in a given time period can only depend on uncertain parameters realized in previous time periods. Also, for tractability reasons, the decision rules are restricted to be linear functions of the uncertain parameters.

An illustrative example and a real-world industrial air separation case demonstrate the capability of the proposed model. The case studies show that significant financial benefits can be achieved by providing interruptible load (1.8% worst-case cost savings in the industrial test case). The results further demonstrate the value of recourse as the cost savings increase with the extent of recourse, which in this case corresponds to the number of uncertain parameters in preceding time periods considered in the decision rules. However, this flexibility in the recourse comes at the cost of increased model size and computation time, which is shown in a brief computational study. Considering reasonable extent of recourse, which results in a solution sufficiently close to the solution with maximum recourse, the industrial test case can be solved within two hours.

Moreover, contrary to results in the literature indicating that demand response is more effective in plants with lower utilization, we have found that this is not true when interruptible load is provided. Here, the largest cost savings are achieved at a high, yet not maximum level of plant utilization. The explanation is that higher plant utilization allows larger amount of interruptible load to be provided, yet some flexibility is still required for the implementation of effective recourse.

**Chapter 7** Industrial DSM should also to be considered at the supply chain level since electricity prices can vary significantly from one location to another, which motivates load shifting not only across different time periods but also across different production plants. In order to consider synergies at the supply chain level,
production and distribution operations have to be coordinated in an efficient manner. In Chapter 7, we have introduced the multiscale production routing problem (MPRP), which considers the simultaneous optimization of production, inventory, distribution, and routing decisions in multicommodity supply chains with power-intensive production facilities.

The proposed MILP model involves two different time grids. Production scheduling is modeled using the fine grid in order to incorporate time-sensitive prices and constraints on operational transitions, while vehicle routing is considered in each time period of the coarse grid. In order to solve MPRP instances of industrially relevant sizes, we have developed an iterative MILP-based heuristic solution method. At each iteration of the proposed algorithm, a restricted MPRP model considering a subset of all possible routes is solved, and the set of candidate routes is updated based on the solutions obtained in previous iterations.

In an extensive computational study, we have considered 50 instances of various sizes. The proposed algorithm is compared with a standard two-phase heuristic approach and a solution strategy involving a one-time heuristic pre-generation of candidate routes. Although the proposed solution method does not guarantee convergence to the optimal solution, the results of the computational study show that it finds high-quality solutions in reasonable computation times (within one hour) and significantly outperforms the other solution approaches in large instances. Similar results are achieved in an industrial case study, which considers a real-world industrial gas supply chain with 2 plants, approximately 240 customers, 20 vehicles, and a planning horizon of 4 weeks, resulting in 168 time periods on the fine grid and 56 time periods on the coarse grid.

Chapter 8 Just like robust optimization, which has been applied in Chapters 5 and 6, flexibility analysis is an approach to solving optimization problems under uncertainty. In Chapter 8, we have demonstrated the connection between flexibility analysis and robust optimization for linear systems; both make use of polyhedral uncertainty sets and the worst-case approach to guarantee feasibility. A comprehensive literature review shows that these two approaches have been developed independently from each other in different research communities at different times. While the major theoretical contributions in flexibility analysis were made in the PSE community in the 1980s, the era of robust optimization in the OR community started in the late 1990s.

To systematically establish the link between flexibility analysis and robust opti-
mization, we have considered the three classical problems from flexibility analysis: the flexibility test problem, the flexibility index problem, and design under uncertainty with flexibility constraints. For LPs with a given general structure, two new solution approaches have been proposed, where the first derives duality-based reformulations of the traditional active-set MILP formulations, and the second applies the concept of affinely adjustable robust optimization (AARO). The concepts that form the theoretical basis for the three different approaches—traditional flexibility analysis (TFA), duality-based flexibility analysis (DFA), and AARO—have been compared. It has become clear that AARO can be seen as a special case of flexibility analysis, however with a duality-based approach to solving the problems. We have shown that in general, AARO is more restrictive and therefore may be overly conservative.

The three different approaches have been applied to three numerical examples, verifying some of the theoretical properties of the proposed formulations. The results further show that DFA and AARO may be computationally more efficient than TFA. The DFA models exhibit a better computational performance because of the tightness of the MILP formulations. In the case of AARO, the main advantage is that only LPs need to be solved; furthermore, no iterative procedure is required for solving the design under uncertainty problem.

9.2. Research Contributions

The main contributions of this thesis are the following:

1. Introduced the concept of Convex Region Surrogate for approximating a nonlinear and nonconvex process model with a union of convex regions in the form of polytopes and region-specific linear cost functions. As shown, a CRS model can be conveniently represented by a set of mixed-integer linear constraints.

2. Developed an algorithm for the construction of CRS models based on given data. The proposed algorithm, which involves solving a number of optimization problems in an iterative framework, has been successfully applied to several data sets including an industrial test case with real process data.

3. Developed a computationally efficient discrete-time MILP model that optimizes the scheduling of process networks where each process is represented by a CRS model. Constraints on operational transitions are formulated by
applying the concept of operating modes, and a block contract formulation is proposed that allows the modeling of various power contract structures.

4. Formulated a two-stage MILP stochastic programming model that considers the simultaneous optimization of production scheduling and electricity procurement under uncertainty in spot electricity price and product demand. CVaR has been incorporated into the model as a measure of risk.

5. Applied scenario reduction and multicut Benders decomposition to solve large instances of the two-stage MILP stochastic program. By doing so, significant reduction in computation time has been achieved compared with directly solving the full-space model.

6. Assessed the operational benefit of an integrated ASU-CES plant by applying an MILP scheduling model that incorporates the capability of recovering energy from CES for internal use, for selling it back to the electricity market, or for providing operating reserve.

7. Applied a robust optimization approach to guarantee feasibility in the integrated ASU-CES model for any possible realization of the reserve demand uncertainty; proposed a budget uncertainty set whose size can be changed in order to adjust the level of conservatism.

8. Developed a multistage adjustable robust MILP model for the scheduling of power-intensive processes providing interruptible load. In the proposed model, recourse decisions are incorporated in the form of linear decision rules, which allows a tractable reformulation of the problem.

9. Proposed a discrete-time MILP model with two time grids for the multiscale production routing problem, which simultaneously optimizes production scheduling and distribution planning. The fine time grid is used to formulate detailed operational constraints, while routing is considered in each time period of the coarse grid in order to guarantee feasible plant-to-customer allocation decisions.

10. Developed an iterative MILP-based heuristic solution method for the MPRP. The proposed algorithm obtains high-quality solutions for MPRPs of industrially relevant sizes in reasonable computation times (one hour), and it is proven to outperform alternative heuristic solution strategies that are considered in practice.
11. Formalized the relation between flexibility analysis and robust optimization for linear systems, showed the conceptual similarities and differences between these two approaches.

12. Proposed for linear systems new efficient formulations for the three classical flexibility analysis problems—flexibility test, flexibility index, and design under uncertainty—based on LP duality and the AARO approach.

**Journal Papers**  The thesis work has led to the following full-length journal papers (listed in the order in which the content appears in this thesis):


**Book Chapters**  Furthermore, we have contributed the following book chapters:


**9.3. Directions for Future Work**

In the following, we provide recommendations for future work, which can be grouped into the following five areas: (1) construction and validation of CRS models, (2) industrial DSM extensions, (3) multiscale integrated optimization, (4) optimization under uncertainty, and (5) extension to nonlinear models.

**9.3.1. Construction and Validation of CRS Models**

Some limitations of the proposed algorithm for the construction of CRS models are listed in Section 2.9. The main concerns are regarding the computational complexity of the algorithm and the accuracy of the resulting CRS models. In the current framework, the algorithm works with a given fixed set of data points. Alternatively, we could sample data points as the algorithm proceeds, and only use data that are useful in the current step of the algorithm. We conjecture that an effective adaptive sampling method can address many of the raised issues, in particular those described in the following:
9. Conclusions

- While Phase 1 requires data points from the interior of the feasible region for the linear fit of the cost function, we only need data points close to the boundary of the feasible region in Phase 2. Therefore, one could ignore most points in the interior of the region in Phase 2 and instead sample more data points at the boundary to obtain a more accurate approximation.

- If the gap between two regions created in Phase 1 is too large, additional data points can be sampled in the empty space between the regions in order to reduce the gap.

- If the simple connectedness assumption does not hold, adaptive sampling can be applied to detect infeasible spaces (“holes”) within the current approximation of the feasible region.

However, for the adaptive sampling framework to be effective, the existing algorithm has to be modified in order to include an efficient update procedure that does not require rerunning the entire algorithm every time new data points are included. Such an update procedure could be further considered in online applications where the surrogate models are updated with new process data in real time.

In this work, we have focused on the generation of CRS models; however, we have not applied systematic model validation procedures to assess the accuracy of the resulting surrogate models. Statistical accuracy measures could be obtained by sampling more data and comparing them with the predictions of the surrogate model. In the same context, cross-validation could be considered to mitigate the effect of overfitting. Also, the proposed algorithm should be tested on more general applications.

9.3.2. Industrial DSM Extensions

In this work, we have considered several problems in industrial DSM, and there are many other DSM applications that can be addressed using similar methodologies. In the following, we list some of these extensions:

- Purchasing electricity from the day-ahead market involves participating in a bidding process where the electricity consumer is required to submit a price-quantity curve, which indicates at which price the consumer is willing to purchase which quantity of electricity. Zhang & Hug (2015) apply a stochastic programming approach to address this problem, but have only considered a relatively simple model of an aluminum smelter. It would be interesting to
apply a similar approach to more complex power-intensive processes using the models developed in this work.

• In Chapter 5 and 6, we have considered the provision of upward operating reserve, which is needed when the electricity supply is less than the demand. We could also encounter the situation in which the supply exceeds the demand, in which case downward operating reserve is required. Power generation facilities can provide downward reserve by reducing their power output, whereas electricity consumers can provide downward reserve by increasing their power consumption. This creates the interesting opportunity for an electricity consumer to get rewarded for consuming electricity; however, sufficient flexibility has to be given in order to guarantee feasible dispatch, which can be achieved by applying a similar modeling approach as in Chapter 6.

• Onsite power generation and energy storage capacities, which are often available at large integrated sites, increase the operational flexibility for DSM. These integrated energy systems should be studied more extensively. The problem becomes especially interesting if power generation from intermittent renewable energy sources is involved, which increases the variability in power output and the level of uncertainty in the system.

• Synergies are expected from collaboration between multiple DSM participants, which could be a network of multiple plants as considered in Chapter 7. It could also be a joint effort of two different companies that operate interrelated power-intensive processes; a typical example is the case of a steel plant and an air separation plant that supplies oxygen to the steel plant. It would be interesting to investigate these potential synergies in an integrated modeling framework.

• Following the same line of thought, it is also clear that collaboration between utility and consumer, or to take it a step further, the grid-wide optimization involving all participants can result in significant benefits for everyone. Here, however, besides the size and complexity of the problem, one also has to consider conflicting objectives among all decision makers, which may require multi-level optimization or a game theoretic approach.
9.3.3. Multiscale Integrated Optimization

The latter DSM problems listed in the previous subsection involve the integration of multiple entities or decision makers. We now consider extensions involving the integration of multiple levels of decision-making. To capture the effect of time-sensitive prices, we usually have to consider an hourly time resolution and a time horizon with the length of at least a day or a week; hence, every integrated problem will require a scheduling component. In the following, we list some integrated DSM problems that can be seen as direct extensions of the work presented in this thesis:

• In Chapter 4, we have considered short-term production scheduling and electricity procurement, where the power contracts are assumed to be given. However, these power contracts typically have to be signed weeks, months, or even years in advance; hence, to decide which contracts to choose in the first place, a long-term electricity procurement problem has to be solved. The electricity procurement decisions made at the planning level have a profound impact on the scheduling decisions; thus, these two problems have to be solved simultaneously.

• The results from Chapter 5 show the potential economic benefit of an integrated ASU-CES system at an operational level; however, we have not considered capital expenses in our analysis. The next step would be to consider the optimal design of an ASU-CES plant, where the challenge lies in the integration of design and operational decisions.

• Recently, the integration of scheduling and control has received increased attention in the PSE community (Nie et al., 2012; Baldea & Harjunkoski, 2014). In the DSM context, this can be important in order to guarantee feasible transitions between different operating points. Also, one could consider providing regulation services. Like interruptible load, regulation is a type of ancillary service that is used to balance supply and demand in the grid; however, regulation is used for frequency control and has to be dispatched within seconds, which requires considering dynamics at the control level.

• As shown in Chapter 7, the integration of production and distribution decisions in supply chains with power-intensive production facilities can lead to significant economic benefits. However, the challenge lies in the computational complexity of solving large-scale instances, which has led to the devel-
9. Conclusions

Development of the MILP-based heuristic method proposed in this work. The disadvantage of the heuristic solution method is that convergence to the optimal solution cannot be guaranteed. It would be interesting to develop efficient exact solution methods for the MPRP, although at this point, it is unlikely that an exact method would be able to solve industrial-scale instances.

- Alternatively, one could consider a special case of the MPRP, namely the case in which we have fixed orders instead of a vendor-managed inventory system. Using Lagrangean decomposition (Guignard, 2003), this problem can be easily decomposed by dualizing the constraints linking the production and distribution parts of the model. Then, the production scheduling problem can be solved for each plant independently, while the distribution planning problem decomposes into independent vehicle routing problems (VRPs), one for each product, plant, and time period. The VRPs can then be solved exactly by, for example, applying well-known branch-and-cut algorithms.

- Many solution methods that have been proposed for the classical production routing problem are metaheuristics, which cannot be directly applied to the MPRP because they cannot accommodate the complex constraints on the production side. However, since metaheuristics are often the state-of-the-art solution methods for solving large-scale routing problems, it would be interesting to see if such an algorithm can be developed for the MPRP and how it would compare with our proposed heuristic.

9.3.4. Optimization Under Uncertainty

Optimization under uncertainty remains a major challenge in industrial DSM. We recommend the following extensions of our uncertainty-related work for further consideration:

- In Chapter 4, the multistage integrated scheduling problem has been approximated by a two-stage MILP stochastic programming model. In this case, the two-stage approximation is acceptable because we consider a relatively short time horizon. However, in many cases, e.g. in the long-term electricity procurement problem, a multistage modeling framework should be explored since it is likely to provide a much more realistic representation of the decision-making process; however, solving such large-scale multistage MILP models will be a major challenge.
• In our application, multicut Benders decomposition has proven to be very efficient in solving large-scale problems. The drawback of Benders decomposition is that it cannot be applied if the subproblems are nonconvex, which would be the case if the model contained integer recourse variables. In such a case, which is likely to occur in long-term strategic planning problems, other decomposition approaches, such as Lagrangean decomposition, bilevel decomposition (Iyer & Grossmann, 1998), and special branch-and-bound algorithms, such as the one proposed by Ahmed et al. (2004), should be considered.

• The provision of interruptible load has been considered in Chapter 6, where we have shown the essential role that recourse plays in this application. Here, we have only considered continuous recourse. In practice, however, we may be able to provide significantly more load reduction by shutting down pieces of equipment or the entire plant. Starting up and shutting down equipment are discrete decisions, which cannot be incorporated as recourse in the current framework. Recently, various methods have been proposed for robust optimization with integer recourse (Bertsimas & Georghiou, 2015; Hanasunto et al., 2015), which could be explored in our DSM applications.

• In Chapter 8, we have shown that the AARO approach is generally more restrictive than traditional flexibility analysis; however, computational experiments have shown that for linear systems, the different approaches often obtain the same results. It remains to be shown if we can derive more general conditions under which the two formulations are guaranteed to achieve the same solution.

9.3.5. Extension to Nonlinear Models

In this work, we have only applied linear models. A natural extension is the consideration of nonlinear models, which may be required for the accurate representation of chemical processes that exhibit highly nonlinear behavior. Naturally, many new challenges arise with nonlinear models:

• By incorporating nonlinearities, one will arrive at large-scale MINLPs when considering similar applications as the ones presented in this thesis. Solving MINLPs is a major challenge. Highly efficient state-of-the-art MILP solvers like CPLEX as well as many decomposition approaches, such as the Benders
decomposition scheme applied in Chapter 4, can no longer be applied. Instead, we would have to resort to MINLP solvers, such as DICOPT, decomposition strategies for MINLPs, such as generalized Benders decomposition (Geoffrion, 1972), or heuristic solution methods. Nonconvexities further increase the computational complexity since finding the global optimum now requires the use of global solvers, such as BARON (Tawarmalani & Sahinidis, 2005), SCIP (Achterberg, 2009), and ANTIGONE (Misener & Floudas, 2014).

- Most concepts and techniques applied in robust optimization, such as the formulation of robust counterparts, rely on linear models. How to model uncertainty in the presence of general nonlinear uncertainty sets or nonlinear relationships involving uncertain parameters within the framework of robust optimization remains an open question. To tackle this problem, it would be interesting to see if one could leverage ideas from flexibility analysis, which can deal with nonlinear models while applying a similar robustness concept.
A. **Alternative Formulation for CRS Model**

Instead of expressing the feasible region of a CRS model as the union of feasible convex regions, we can also formulate it as the difference of the convex hull and the union of infeasible convex regions. In our example, as shown in Figure A.1, the infeasible convex regions are the two empty polytopes.

![Figure A.1: CRS expressed as the difference of the convex hull around all data points and the union of infeasible (empty) convex regions.](image)

A polytope can be seen as an intersection of a set of half-spaces of which each is bounded by the hyperplane containing the corresponding facet of the polytope. A point in the polytope, $x$, is then a solution of the set of constraints $a_h^T x \leq b_h \ \forall \ h \in H$, where $H$ is the set of half-spaces. This is commonly referred to as an H-representation of a polyhedron. By applying the H-representation to the infeasible regions, we can express a feasible point in the CRS model, $x$, as a solution to the following set of constraints:

\begin{align}
  x &= \sum_j \lambda_j v_j \quad \forall \ j \in V \quad \text{(A.1a)} \\
  \sum_{j \in V} \lambda_j &= 1 \quad \text{(A.1b)} \\
  0 \leq \lambda_j \leq 1 \quad \forall \ j \in V \quad \text{(A.1c)} \\
  a_{rh}^T x &\geq b_{rh} + \bar{e} - M(1 - z_{rh}) \quad \forall \ r \in R, \ h \in H_r \quad \text{(A.1d)}
\end{align}
A. Alternative Formulation for CRS Model

\[ \sum_{h \in H_r} z_{rh} \geq 1 \quad \forall r \in \bar{R} \]  \hspace{1cm} (A.1e)

\[ z_{rh} \in \{0, 1\} \quad \forall r \in \bar{R}, h \in H_r \]  \hspace{1cm} (A.1f)

where \( V \) is the set of vertices of the convex hull, \( \bar{R} \) is the set of infeasible convex regions, \( H_r \) is the set of half-spaces associated with region \( r \), and \( \epsilon \) is a small margin parameter. Eqs. (A.1a)–(A.1c) state that \( x \) is a point inside the convex hull, while Eqs. (A.1d)–(A.1f) enforce that \( x \) is not a point in the interior of any of the infeasible regions. The binary variable \( z_{rh} \) equals 1 if the constraint \( a_{rh}^T x \leq b_{rh} \) is violated. Note that this approach could also be used to consider “holes” in the feasible region.

Formulation (A.1) is not entirely equivalent to Eqs. (2.3b)–(2.3g) since points on facets shared by feasible and infeasible regions will not be feasible in (A.1). Moreover, the big-M parameter \( M \) in (A.1d) typically leads to weak LP relaxations which will likely make the alternative formulation less efficient if the numbers of feasible and infeasible convex regions are similar.
B. DATA FOR CRS ILLUSTRATIVE EXAMPLE

For the illustrative example, we have constructed the set of 100 data points shown in Table B.1. Each data point consists of a 2-dimensional parameter vector \( a \) and a cost value \( g \). The cost values are calculated from the parameter values using two different linear correlations, for which the corresponding constants and coefficients are listed in Table B.2. The cost values for the first 52 points are generated from the first linear correlation, while the cost values for the remaining points are calculated by applying the second linear correlation.

Table B.1: Complete set of data for the illustrative example.

| \(a_1\) | 4   | 14  | 6   | 11  | 5   | 13  | 3   | 4   | 6   | 8   | 10  | 12  | 14  | 15  | 3.5 | 5   | 7   | 9   | 11  | 13  |
|--------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| \(a_2\) | 7   | 7   | 8   | 8   | 8.5 | 8.5 | 9   | 9   | 9   | 9   | 9   | 9   | 10  | 10  | 10  | 10  | 10  | 10  | 10  |
| \(g\)  | 39  | 59  | 46  | 56  | 45.5| 61.5| 43  | 45  | 49  | 53  | 57  | 61  | 65  | 67  | 67  | 67  | 50  | 54  | 58  | 62  | 66  |
| \(a_1\) | 14.5| 2   | 4   | 6   | 8   | 10  | 12  | 14  | 16  | 3   | 5   | 7   | 9   | 11  | 13  | 15  | 3   | 4   | 6   | 8   |
| \(a_2\) | 10  | 11  | 11  | 11  | 11  | 11  | 11  | 11  | 11  | 11  | 11  | 12  | 12  | 12  | 12  | 12  | 12  | 12  | 12  | 13  | 13  | 13  |
| \(g\)  | 69  | 47  | 51  | 55  | 59  | 63  | 67  | 71  | 75  | 52  | 56  | 60  | 64  | 68  | 72  | 76  | 55  | 57  | 57  | 61  | 65  |
| \(a_1\) | 10  | 12  | 14  | 5   | 11  | 13  | 15  | 4   | 12  | 14  | 13  | 14  | 14  | 14  | 14  | 15  | 15  | 16  | 17  | 2   | 2   | 2   | 2   | 3   | 3   |
| \(a_2\) | 13  | 13  | 13  | 14  | 14  | 14  | 14  | 14  | 15  | 15  | 15  | 16  | 17  | 2   | 2   | 2   | 2   | 2   | 3   | 3   | 3   |
| \(g\)  | 69  | 73  | 77  | 62  | 74  | 78  | 82  | 63  | 79  | 83  | 84  | 89  | 320 | 380 | 440 | 500 | 560 | 400 | 460 | 520 |
| \(a_1\) | 11  | 3   | 4.5 | 6   | 8   | 10  | 11.5| 13  | 3.5 | 5   | 7   | 9   | 11  | 12.5| 2   | 4   | 6   | 8   | 10  | 12  | 12  |
| \(a_2\) | 3   | 4   | 4   | 4   | 4   | 5   | 5   | 5   | 5   | 5   | 5   | 5   | 6   | 6   | 6   | 6   | 6   | 6   | 6   | 6   |
| \(g\)  | 580 | 390 | 435 | 480 | 540 | 600 | 645 | 690 | 455 | 500 | 560 | 620 | 680 | 725 | 460 | 520 | 580 | 640 | 700 | 760 |
| \(a_1\) | 14  | 6   | 8   | 7   | 10  | 9   | 12  | 7   | 10  | 14  | 8   | 12  | 9   | 11  | 13  | 14  | 8   | 10  | 12  | 12  |
| \(a_2\) | 6   | 16  | 16.5| 17  | 17  | 17.5| 17.5| 18  | 18  | 18  | 18  | 18  | 18.5| 19  | 19  | 19  | 19  | 19  | 19  | 20  | 20  |
| \(g\)  | 820 | 1080| 1165| 1160| 1250| 1245| 1335| 1210| 1300| 1420| 1265| 1385| 1320| 1380| 1440| 1470| 1340| 1400| 1460| 1520|

Table B.2: Cost constants and coefficients used in the illustrative example.

<table>
<thead>
<tr>
<th></th>
<th>(b)</th>
<th>(c_1)</th>
<th>(c_2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>First 52 points</td>
<td>10</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Remaining 48 points</td>
<td>100</td>
<td>30</td>
<td>50</td>
</tr>
</tbody>
</table>
C. DERIVATION OF THE ADJUSTABLE ROBUST COUNTERPART

To each constraint in Problem (6.15), we apply the worst-case approach with respect to the uncertainty. As a result, the following bilevel problem is obtained:

\[
\begin{align*}
\min_{t \in T} & \left[ \sum_{m} \sum_{r \in R_m} \alpha_i^E C \left( \delta_{mr} \bar{y}_{mrt} + \sum_i \gamma_{mri} \bar{P}_D \right) + \sum_i \alpha_i^P \bar{C}_i \right] - \alpha_i^IL t \\
+ & \sum_{m} \sum_{r \in R_m} \sum_i \alpha_i^P \left( -p_{mrit} + \sum_{k=t-\zeta_t}^{t-1} p_{mritk} \right) \right] + \sum_i \sum_{k=t-\zeta_t}^{t} \alpha_i^P q_{itk} \right] \\
+ & \max_{w \in W(IL)} \left\{ \sum_{m} \sum_{r \in R_m} \sum_{k=t-\zeta_t}^{t} \alpha_i^P \gamma_{mri} P_{mritk} w_k + \sum_i \sum_{k=t-\zeta_t}^{t} \alpha_i^P q_{itk} w_k \right\} \\
\text{s.t.} & \text{Eqs. (6.1e)--(6.1f), (6.2), (6.4b)--(6.4c), (6.7), (6.13c)--(6.13e), (6.15f)--(6.15k)} \\
\sum_{i} a_{mrfi} \bar{P}_D + \min_{w \in W(IL)} \left\{ \sum_{i} a_{mrfi} \sum_{k=t-\zeta_t}^{t} p_{mritk} w_k \right\} & \geq b_{mrf} \bar{y}_{mrt} \\
& \forall m, r \in R_m, f \in F_m, t \in T \\
\bar{P}_D + & \max_{w \in W(IL)} \left\{ \sum_{k=t-\zeta_t}^{t} p_{mritk} w_k \right\} \leq \bar{P}_D_{mri} \bar{y}_{mrt} \forall m, r \in R_m, t \in T \\
\bar{P}_D + & \min_{w \in W(IL)} \left\{ \sum_{k=t-\zeta_t}^{t} p_{mritk} w_k \right\} \geq 0 \forall m, r \in R_m, t \in T \\
IV_{i,0} + & \sum_{k=1}^{t} \left( \sum_{m} \sum_{r \in R_m} \bar{P}_D_{mrik} + \bar{P}_C \right) - D_{ik} \right) \\
+ & \max_{w \in W(IL)} \left\{ \sum_{k=1}^{t} \left( \sum_{m} \sum_{r \in R_m} \sum_{l=k-\zeta_k}^{k} p_{mrikl} w_l + \sum_{l=k-\zeta_k}^{k} \eta_{ikl} w_l \right) \right\} \leq IV_{i,0} \forall i, t \in T \\
IV_{i,0} + & \sum_{k=1}^{t} \left( \sum_{m} \sum_{r \in R_m} \bar{P}_D_{mrik} + \bar{P}_C \right) - D_{ik} \right)
\end{align*}
\]

(C.1a)
Tables C.1 and C.2 list the lower-level problems and the corresponding dual formulations, respectively. Due to strong duality, the bilevel problem can be reformulated into a single-level problem by substituting the dual formulations into Eqs. (C.1), which yields the ARC given by Eqs. (6.17).

**Table C.1: List of lower-level problems**

<table>
<thead>
<tr>
<th>A</th>
<th>$\max_w \left{ \sum_{k \in T} \left( \sum_{i \in R_m} \left( \sum_{k = \zeta_i}^t a_{EC}^{\gamma_{mri}} p_{mritk} w_k + \sum_{t \in k - \zeta_i} a_{PC}^{\gamma_{mri}} q_{ikt} w_k \right) \right) \right}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>$\min_w \left{ \sum_{k = t - \zeta} p_{mritk} w_k : \sum_{k = t - \zeta} w_k \leq \Gamma_t, 0 \leq w_k \leq 1 \forall k \in K_{t - \zeta}, t \right}$</td>
</tr>
<tr>
<td>C</td>
<td>$\max_w \left{ \sum_{k = t - \zeta} p_{mritk} w_k : \sum_{k = t - \zeta} w_k \leq \Gamma_t, 0 \leq w_k \leq 1 \forall k \in K_{t - \zeta}, t \right}$</td>
</tr>
<tr>
<td>D</td>
<td>$\min_w \left{ \sum_{k = t - \zeta} p_{mritk} w_k : \sum_{k = t - \zeta} w_k \leq \Gamma_t, 0 \leq w_k \leq 1 \forall k \in K_{t - \zeta}, t \right}$</td>
</tr>
<tr>
<td>E</td>
<td>$\max_w \left{ \sum_{k = 1}^t \left( \sum_{i \in R_m} \left( \sum_{k = \zeta_i}^k p_{mritk} w_l + \sum_{t \in k - \zeta_i} q_{ikt} w_l \right) \right) \right}$</td>
</tr>
<tr>
<td>F</td>
<td>$\min_w \left{ \sum_{k = 1}^t \left( \sum_{i \in R_m} \left( \sum_{k = \zeta_i}^k p_{mritk} + q_{ikt} \right) \right) w_k : \sum_{k = 1}^t w_k \leq \Gamma_t, 0 \leq w_k \leq 1 \forall k \in K_{1}, t \right}$</td>
</tr>
<tr>
<td>G</td>
<td>$\max_w \left{ \sum_{k = t - \zeta}^t \sum_{m \in R_m} \left( \sum_{i \in R_m} \sum_{t \in k - \zeta_i} \gamma_{mri} p_{mritk} w_k + I L_t w_l \right) \right}$</td>
</tr>
</tbody>
</table>

279
Table C.2: Dual formulations of lower-level problems

\[
\begin{align*}
\text{A} & \quad \min_{u^A, s^A} \left\{ \Gamma_t u^A + \sum_{k=1}^{i} s_k^A : u^A + s_k^A \geq \sum_{k=1}^{k_{\text{max}}} \left( \sum_{i=1}^{i_{\text{max}}} \alpha_i \gamma_m p_m r_i k + \alpha_i p_{\text{C}} q_{i k} \right) \right\} \quad \forall k \in K_{1,i}, \\
& \quad u^A \geq 0, s_k^A \geq 0 \quad \forall k \in K_{1,i} \\
\text{B} & \quad \max_{u^B, s^B} \left\{ \Gamma_t u^B + \sum_{k=t-\Gamma_{t}}^{t} s_{mrfk}^B : u^B + s_{mrfk}^B \leq \sum_{i} a_{mrfi} p_{mrfk} \right\} \quad \forall k \in K_{t-\Gamma_{t}, t}, \\
& \quad u^B \leq 0, s_{mrfk}^B \leq 0 \quad \forall k \in K_{t-\Gamma_{t}, t} \\
\text{C} & \quad \min_{u^C, s^C} \left\{ \Gamma_t u^C + \sum_{k=t-\Gamma_{t}}^{t} s_{mritk}^C : u^C + s_{mritk}^C \geq p_{mritk} \right\} \quad \forall k \in K_{t-\Gamma_{t}, t}, \\
& \quad u^C \geq 0, s_{mritk}^C \geq 0 \quad \forall k \in K_{t-\Gamma_{t}, t} \\
\text{D} & \quad \max_{u^D, s^D} \left\{ \Gamma_t u^D + \sum_{k=t-\Gamma_{t}}^{t} s_{mritk}^D : u^D + s_{mritk}^D \leq p_{mritk} \right\} \quad \forall k \in K_{t-\Gamma_{t}, t}, \\
& \quad u^D \leq 0, s_{mritk}^D \leq 0 \quad \forall k \in K_{t-\Gamma_{t}, t} \\
\text{E} & \quad \min_{u^E, s^E} \left\{ \Gamma_t u^E + \sum_{k=1}^{k_{\text{max}}} s_{\text{MIL}}^E : u^E + s_{\text{MIL}}^E \geq \sum_{l=1}^{l_{\text{max}}} \left( \sum_{i=1}^{i_{\text{max}}} p_{mrfi} + q_{l k} \right) \right\} \quad \forall k \in K_{1,t}, \\
& \quad u^E \geq 0, s_{\text{MIL}}^E \geq 0 \quad \forall k \in K_{1,t} \\
\text{F} & \quad \max_{u^F, s^F} \left\{ \Gamma_t u^F + \sum_{k=1}^{k_{\text{max}}} s_{\text{MIL}}^F : u^F + s_{\text{MIL}}^F \leq \sum_{l=1}^{l_{\text{max}}} \left( \sum_{i=1}^{i_{\text{max}}} p_{mrfi} + q_{l k} \right) \right\} \quad \forall k \in K_{1,t}, \\
& \quad u^F \leq 0, s_{\text{MIL}}^F \leq 0 \quad \forall k \in K_{1,t} \\
\text{G} & \quad \min_{u^G, s^G} \left\{ \Gamma_t u^G + \sum_{k=t-\Gamma_{t}}^{t} s_{tk}^G : u^G + s_{tk}^G \geq \sum_{m} \sum_{r \in R_m} \gamma_m p_{mritk} \right\} \quad \forall k \in K_{t-\Gamma_{t}, t-1}, \\
& \quad u^G + s_{tk}^G \geq \sum_{m} \sum_{r \in R_m} \gamma_m p_{mritk} + IL_t, \\
& \quad u^G \geq 0, s_{tk}^G \geq 0 \quad \forall k \in K_{t-\Gamma_{t}, t} 
\end{align*}
\]
D. DERIVATION OF THE STATIC ROBUST COUNTERPART

In the following, we present the derivation of the robust counterpart when applying the traditional static robust optimization approach without recourse to the specific case in which the plant only produces one single product. Since we consider the single-product case, the product index $i$ is omitted for all variables.

In the robust model without recourse, we assume that when load reduction is requested, the electricity consumption due to production is decreased such that the required amount is exactly met. Therefore, inequality (6.7) becomes an equality:

$$\sum_{m} \sum_{r \in R_m} \gamma_{mr} \overline{PD}_{mrt} = -IL_t w_t \quad \forall t \in \overline{T}$$ \hspace{1cm} (D.1)

which can be replaced by

$$\overline{PD}_{mrt} = -\frac{IL_t w_t}{\gamma_{mr}} \overline{y}_{mrt} \quad \forall m, r \in R_m, t \in \overline{T}$$ \hspace{1cm} (D.2)

since for each $t \in \overline{T}$, only one $\overline{y}_{mrt}$ equals 1 and allows the corresponding $\overline{PD}_{mrt}$ to be nonzero. To avoid bilinear terms, we further replace Eq. (D.2) by the following equations:

$$\overline{PD}_{mrt} = -\frac{IL_{mrt} w_t}{\gamma_{mr}} \overline{y}_{mrt} \quad \forall m, r \in R_m, t \in \overline{T}$$ \hspace{1cm} (D.3a)

$$IL^{\min}_{mrt} \leq \overline{IL}_{mrt} \leq IL^{\max}_{mrt} \quad \forall m, r \in R_m, t \in \overline{T}$$ \hspace{1cm} (D.3b)

which allow the formulation of the following robust uncertain problem:

$$\min \sum_{t \in T} \left[ \sum_{m \in R_m} \sum_{r \in R_m} \alpha^EC_t (\delta_{mr} \overline{y}_{mrt} + \gamma_{mr} \overline{PD}_{mrt}) + \alpha^PC_t \overline{PC}_{t} - \alpha^{IL}_{t} \sum_{m \in R_m} \sum_{r \in R_m} \overline{IL}_{mrt} \right]$$

$$+ \max_{w \in W(IL)} \left\{ \sum_{t \in T} \sum_{m \in R_m} \sum_{r \in R_m} \alpha^EC_t \gamma_{mr} \left( -\frac{\overline{IL}_{mrt} w_t}{\gamma_{mr}} \right) \right\}$$ \hspace{1cm} (D.4a)
D. Derivation of the Static Robust Counterpart

s.t. Eqs. (6.1e)–(6.1f), (6.2), (6.4b)–(6.4c), (6.13c)–(6.13e), (6.15f)–(6.15k), (D.3b)
\[ IL_{\text{min}}^x x_t \leq \overline{T}L_{\text{mrt}} \leq IL_{\text{max}}^x x_t \quad \forall \ m, r \in R_m, t \in \overline{T} \]  
\[ a_{\text{mrf}} \overline{PD}_{\text{mrt}} + \min_{w \in W(IL)} \left\{ a_{\text{mrf}} \left( \frac{\overline{T}L_{\text{mrt}} w_t}{\gamma_{\text{mr}}} \right) \right\} \geq b_{\text{mrf}} \overline{y}_{\text{mrt}} \quad \forall \ m, r \in R_m, f \in F_m, t \in T \]  
\[ \overline{PD}_{\text{mrt}} + \max_{w \in W(IL)} \left\{ -\frac{\overline{T}L_{\text{mrt}} w_t}{\gamma_{\text{mr}}} \right\} \leq PD_{\text{mrt}}^{\text{max}} \overline{y}_{\text{mrt}} \quad \forall \ m, r \in R_m, t \in T \]  
\[ \overline{PD}_{\text{mrt}} + \min_{w \in W(IL)} \left\{ -\frac{\overline{T}L_{\text{mrt}} w_t}{\gamma_{\text{mr}}} \right\} \geq 0 \quad \forall \ m, r \in R_m, t \in T \]  
\[ IV_0 + \sum_{k=1}^{t} \left( \sum_{m \in R_m} \gamma_{\text{mr}} \overline{PD}_{\text{mrk}} + \overline{PC}_k - D_k \right) \]  
\[ + \max_{w \in W(IL)} \left\{ \gamma_{\text{mr}} \overline{PD}_{\text{mrk}} + \overline{PC}_k - D_k \right\} \leq IV_{\text{max}} \quad \forall \ t \in \overline{T} \]  
\[ IV_0 + \sum_{k=1}^{t} \left( \sum_{m \in R_m} \gamma_{\text{mr}} \overline{PD}_{\text{mrk}} + \overline{PC}_k - D_k \right) \]  
\[ + \min_{w \in W(IL)} \left\{ \gamma_{\text{mr}} \overline{PD}_{\text{mrk}} + \overline{PC}_k - D_k \right\} \geq IV_{\text{min}} \quad \forall \ t \in T \]

The same approach for the derivation of the ARC can be applied to transform this bilevel problem into a single-level problem. The resulting robust counterpart is as follows:

\[ \min \sum_{t \in T} \left[ \sum_{m \in R_m} \alpha_{t}^{EC} (\delta_{\text{mrf}} \overline{y}_{\text{mrt}} + \gamma_{\text{mr}} \overline{PD}_{\text{mrt}}) + \alpha_{t}^{PC} \overline{PC}_t - \alpha_{t}^{IL} \sum_{m \in R_m} \overline{T}L_{\text{mrt}} \right] \]

s.t. Eqs. (6.1e)–(6.1f), (6.2), (6.4b)–(6.4c), (6.13c)–(6.13e), (6.15f)–(6.15k), (D.3b)
\[ IL_{\text{min}}^x x_t \leq \overline{T}L_{\text{mrt}} \leq IL_{\text{max}}^x x_t \quad \forall \ m, r \in R_m, t \in \overline{T} \]  
\[ a_{\text{mrf}} \left( \overline{PD}_{\text{mrt}} - \frac{\overline{T}L_{\text{mrt}}}{\gamma_{\text{mr}}} \right) \geq b_{\text{mrf}} \overline{y}_{\text{mrt}} \quad \forall \ m, r \in R_m, f \in F_m, t \in \overline{T} \]  
\[ \overline{PD}_{\text{mrt}} \leq PD_{\text{mrt}}^{\text{max}} \overline{y}_{\text{mrt}} \quad \forall \ m, r \in R_m, t \in \overline{T} \]  
\[ \overline{PD}_{\text{mrt}} - \frac{\overline{T}L_{\text{mrt}}}{\gamma_{\text{mr}}} \geq 0 \quad \forall \ m, r \in R_m, t \in \overline{T} \]  
\[ IV_0 + \sum_{k=1}^{t} \left( \sum_{m \in R_m} \overline{PD}_{\text{mrk}} + \overline{PC}_k - D_k \right) \leq IV_{\text{max}} \quad \forall \ t \in \overline{T} \]  
\[ IV_0 + \sum_{k=1}^{t} \left( \sum_{m \in R_m} \overline{PD}_{\text{mrk}} + \overline{PC}_k - D_k \right) + \Gamma_t u_t + \sum_{k=1}^{t} s_{tk} \geq IV_{\text{min}} \quad \forall \ t \in \overline{T} \]  

282
D. DERIVATION OF THE STATIC ROBUST COUNTERPART

\begin{align*}
  u_t + s_{tk} & \leq - \sum_{m \in R_m} \sum_{r \in R_m} \frac{T_{r, mk}}{\gamma_{mk}} & \forall t \in T, k \in K_{1,t} \\
  u_t & \leq 0 & \forall t \in T \\
  s_{tk} & \leq 0 & \forall t \in T, k \in K_{1,t}
\end{align*}

(D.5h) (D.5i) (D.5j)
The routing problem in Phase 2 of Heuristic H1 can be solved for each product $i$, plant $p$, and time period $t \in \mathcal{T}$ independently. For each $(i, p, t)$, we consider plant $p$ and the customers in $C_i$ that receive deliveries from plant $p$ in time period $t$. We create a complete directed graph $G = (N, A)$ where $N = \{0, 1, \ldots, n^c\}$ represents the set of the plant (node 0) and the customers and $A = \{(n, n') : n \in N, n' \in N, n \neq n'\}$ is the set of arcs. We further define the set of customers $\tilde{N} = N \setminus \{0\}$ and the set of arcs between customers $\tilde{A} = \{(n, n') : n \in \tilde{N}, n' \in \tilde{N}, n \neq n'\}$. The fixed order from customer $n$, denoted by $O_n$, is set to $DLict$ (for the corresponding $c$ in the customer set), which is obtained in Phase 1. The time required for visiting node $n'$ from node $n$ is denoted by $\hat{\tau}_{nn'}$, which consists of the travel time from $n$ to $n'$ and the average stay time at one node. Similarly, the corresponding distribution cost, denoted by $\hat{\beta}_{nn'}$, includes the travel cost and the fixed cost for loading and unloading.

For each $(i, p, t)$, we solve the following MILP model of the DCVRP with product purchase:

$$\min \sum_{(n, n') \in \tilde{A}} \hat{\beta}_{nn'} w_{nn'} + \sum_{n \in \tilde{N}} \alpha_n \overline{PC}_n$$

s.t.

$$\sum_{n' \in N, n' \neq n} w_{nn'} = \sum_{n' \in N, n' \neq n} w_{n'n} \quad \forall \ n \in \tilde{N}$$

$$\sum_{n \in \tilde{N}} w_{0,n} \leq L_{ip}$$

$$u_n - u_{n'} + V_i w_{nn'} + (V_i - O_n - O_{n'}) w_{n'n} \leq V_i - O_{n'} \quad \forall \ (n, n') \in \tilde{A}$$

$$O_n + \sum_{n' \in \tilde{N}, n' \neq n} O_{n'} w_{n'n} \leq u_n \quad \forall \ n \in \tilde{N}$$

$$u_n \leq V_i - \sum_{n' \in \tilde{N}, n' \neq n} O_{n'} w_{n'n} \quad \forall \ n \in \tilde{N}$$

$$\bar{u}_n - \bar{u}_{n'} + \overline{\tau}_{n'n} x_{nn'} + (\overline{\tau}_{n'n} - \bar{\tau}_{nn'} - \overline{\tau}_{n'n}) w_{n'n} \leq \overline{\tau}_{n'n} - \bar{\tau}_{nn'}$$

$$\forall \ (n, n') \in \tilde{A}$$
E. DCVRP Formulation

\[ \hat{\tau}_{0,n} + \sum_{n' \in \mathbb{N}, n' \neq n} (\hat{\tau}_{0,n'} + \hat{\tau}_{n'n} - \hat{\tau}_{0,n}) w_{n'n} \leq \bar{u}_n \quad \forall \ n \in \tilde{\mathbb{N}} \]  
\text{(E.1h)}

\[ \bar{u}_n \leq \bar{\tau}_{\text{max}} - \hat{\tau}_{n,0} - \sum_{n' \in \mathbb{N}, n' \neq n} (\hat{\tau}_{n',0} + \hat{\tau}_{nn'} - \hat{\tau}_{n,0}) w_{nn'} \quad \forall \ n \in \tilde{\mathbb{N}} \]  
\text{(E.1i)}

\[ O_n \sum_{n' \in \mathbb{N}, n' \neq n} w_{nn'} + PC_n = O_n \quad \forall \ n \in \tilde{\mathbb{N}} \]  
\text{(E.1j)}

\[ PC_n \geq 0 \quad \forall \ n \in \tilde{\mathbb{N}} \]  
\text{(E.1k)}

\[ w_{nn'} \in \{0, 1\} \quad \forall \ (n, n') \in A \]  
\text{(E.1l)}

where the binary variable \( w_{nn'} \) equals 1 if a vehicle travels from node \( n \) to node \( n' \), and \( PC_n \) denotes the amount of product purchased for customer \( n \). Eq. (E.1a) states the objective function, which consists of the distribution cost and the product purchase cost, with \( \bar{\alpha}_n \) denoting the unit cost for purchasing product for customer \( n \). Eq. (E.1b) represents the vehicle flow conservation constraints, while constraint (E.1c) limits the number vehicles according to the availability at plant \( p \). We adopt the lifted formulation of the Miller-Tucker-Zemlin (MTZ) subtour elimination constraints proposed by Desrochers & Laporte (1991), which are stated in Eqs. (E.1d)–(E.1i). Eqs. (E.1d)–(E.1f) further restrict the amount of product delivered on each trip to the vehicle capacity \( V_i \), while Eqs. (E.1g)–(E.1i) prohibit the selection of trips that take longer than \( \bar{\tau}_{\text{max}} \). Finally, according to Eq. (E.1j), the order quantity for each customer is either fully met by delivery or by purchase.
BIBLIOGRAPHY


286


GAMS Development Corporation (2014). GAMS version 24.2.3.


R Core Team (2014). R: A Language and Environment for Statistical Computing.


