Optimal Multi-scale Capacity Planning for Power-Intensive Continuous Processes under Time-sensitive Electricity Prices and Demand Uncertainty
Part II: Enhanced Hybrid Bi-level Decomposition

Sumit Mitra*, Jose M. Pinto†, Ignacio E. Grossmann*‡

May 22, 2013

Abstract

We describe a hybrid bi-level decomposition scheme that addresses the challenge of solving a large-scale two-stage stochastic programming problem with mixed-integer recourse, which results from a multi-scale capacity planning problem as described in part I of this paper series. The decomposition scheme combines bi-level decomposition with Benders decomposition, and relies on additional strengthening cuts from a Lagrangean-type relaxation and subset-type cuts from structure in the linking constraints between investment and operational variables. The application of the scheme with a parallel implementation to an industrial case study reduces the computational time by two orders of magnitude when compared with the time required for the solution of the full-space model.

1 Introduction

In part I of the paper, we described a model for the integration of operational and strategic decision-making for continuous power-intensive processes under time-sensitive electricity prices and product demand uncertainty. The resulting formulation is a two-stage stochastic programming problem (Birge and Louveaux, 2011), whose deterministic equivalent is a large-scale MILP due to the integration of different time-scales, from hourly decisions on production levels and modes to investments decisions over a horizon of multiple

*Center for Advanced Process Decision-making, Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213

†Praxair, Inc., Danbury, CT

‡Corresponding author. Email address: grossmann@cmu.edu

years. Therefore, the problem is hard to solve and it already has nearly 1 million constraints, 2.4 million variables (of which 221,780 are binary) for the case of 60 scenarios, which result from modeling a ten year horizon with an aggregated time representation and three scenarios per season. At the same time, the problem has a structure that deserves special attention. Similar to other two-stage stochastic programming problems, the problem decomposes into individual operational subproblems once first-stage investment decisions are fixed. However, one major challenge is the large number of binary decision variables in the second stage that originate from detailed scheduling problems.

There are two main decomposition schemes that have been applied to two-stage stochastic programming problems, namely Lagrangean decomposition (Guignard and Kim, 1987; Caroe and Schultz, 1999) and Benders decomposition also known as L-shaped method (Benders, 1962; Geoffrion, 1972; Van Slyke and Wets, 1969).

Lagrangean decomposition applied to stochastic programming problems is a special form of Lagrangean relaxation, which decomposes the original problem into subproblems by duplicating the first-stage investment variables and dualizing the so-called non-anticipativity constraints that enforce the same first-stage investment decisions across all scenarios. The multipliers of the dualized non-anticipativity constraints (the so-called complicating constraints) are iteratively updated with subgradient optimization or cutting planes. Lagrangean decomposition can also be applied for non-convex problems. However, the duality gap that arises can only be closed by using a branch-and-bound enumeration in the full variable space (Karuppiah and Grossmann, 2008).

Benders decomposition solves the original problem by evaluating the second-stage subproblems for different realizations of the complicating variables. The search in the space of the complicating variables is performed with a master problem that collects dual information from the subproblems, which describe the sensitivity of the second-stage decisions with respect to the first-stage decisions. Note that the collection of dual information relies on strong duality and becomes difficult if the problem is non-convex in the second-stage variables. Recently, Li et al. (2011) extend the idea of Benders decomposition to non-convex problems by replacing the original non-convex problem by a convex relaxation and applying Benders decomposition to the relaxation. Sundaramoorthy et al. (2012) apply Li et al.'s algorithm to a two-stochastic programming problem that represents a capacity planning problem in the pharmaceutical industry.

Another decomposition scheme, which in contrast to Benders and Lagrangean decomposition does not rely on dual information, is the so-called bi-level decomposition algorithm (Iyer and Grossmann, 1998). Bi-level decomposition has been used in various applications, ranging from investment planning problems for utility plants (Iyer and Grossmann, 1998), oil fields

(Van den Heever and Grossmann, 1999), and supply chains (You et al., 2010) to planning and scheduling problems (Erdirik-Dogan and Grossmann, 2007). The algorithm is also based on the idea that some decision variables of the problem are complicating variables, e.g. investment decisions in strategic planning problems or assignment variables in planning and scheduling problems.

However, in contrast to Benders decomposition, the master problem is an aggregated problem (AP) that corresponds to a tailored relaxation of the original problem, typically obtained by relaxing 0/1 variables, relaxing some constraints and/or taking linear combinations of them. The aggregated problem (AP) yields an initial bound, normally much tighter than the one obtained from the initial Benders master problem. (AP) is solved alternately with the detailed problem (DP), in which the complicating variables are restricted. Primal cuts are inferred from (DP) and added back to (AP), and the process iterates until the gap between the objective function values of the two problems is within a specified tolerance. Recently, Calfa et al. (2013) as well as Terrazas-Moreno and Grossmann (2011) apply Lagrangean decomposition within bi-level decomposition to the aggregated problem (AP). While both authors can speed up the solution process significantly, their schemes lead to duality gaps due to the application of Lagrangean decomposition.

In this part of the paper, we focus on the development and application of a suitable decomposition strategy for the two-stage multi-scale stochastic programming problem that we described in part I of the paper. We intend to combine the individual strengths of the afore mentioned decomposition schemes. First, the problem statement is reviewed in section 2. The classical bi-level decomposition algorithm is introduced in section 3 and applied to our problem. We derive subset-type cuts for the second-stage value function based on the solution of the detailed problem (DP). In section 4, we explain how a Lagrangean-type relaxation of (DP) can be used to generate good initial bounds, and how Benders decomposition is used to further decompose the aggregated problem (AP). The complete enhanced hybrid bi-level decomposition algorithm is described in section 5. Finally, in section 6, we discuss the parallel implementation of our scheme within the GAMS grid computing environment (Bussieck, Ferris and Meeraus, 2009), and show computational results that demonstrate the impact of our decomposition algorithm for the multi-scale capacity planning problem applied to the industrial case study from part I of this paper.

2 Problem statement

We would like to solve a two-stage stochastic programing problem for the multi-scale capacity planning of a continuous power-intensive process under time-sensitive electricity prices and product demand uncertainty. The scenarios correspond to different demand realizations over the time horizon and the problem has complete recourse since variables for external product purchases with associated cost terms in the objective function are present. The problem we introduced in part I of this paper can be summarized in the following way:

$$
\begin{aligned}
(P) \quad \min \quad & \sum_{t' \in T_{invest}} c_{t'}^T x_{t'} + \sum_{t \in T, s \in S} \tau_{t,s} d_{t,s}^T y_{t,s} && (1)\\
s.t. \quad & \sum_{t' \in T_{invest}} A_{0,t'} x_{t'} \leq b_0 && (2)\\
& \sum_{t' \in T_{invest}, t' \leq t} A_{1,t'} x_{t'} + B_1 y_{t,s} \leq b_1 \quad \forall t \in T, s \in S && (3)\\
& y_{t,s} \in Y_{t,s} \quad \forall t \in T, s \in S && (4)\\
& x_{t'} \in \{0,1\}^n \quad \forall t' \in T_{invest} && (5)
\end{aligned}
$$

The objective function (1) minimizes the sum of capital expenditures ($CAPEX_{t'} = c_{t'}^T x_{t'}$), as defined by equation (23) in part I, and operating expenditures ($OPEX_{t,s} = d_{t,s}^T y_{t,s}$) over a set of seasons $t \in T$ and scenarios $s \in S$ with probabilities $\tau_{t,s}$, as defined by equation (24) in part I. The first-stage variables, $x_{t'}$, are binary and involve decisions on a set of investments (N, $|N| = n$) with fixed capacities, which are allowed in certain time periods, T_{invest} (in our case at the beginning of each year), as described in equation (5). Equation (2) specifies the restrictions on the investment decisions $x_{t'}$, such as constraints (17) and (20) from part I, which allow certain investments (new major equipment and equipment upgrades) to be executed only once over the time horizon. Equation (3) summarizes the linking constraints between investment decisions $x_{t'}$ and operational second-stage decisions $y_{t,s}$, which include constraints (15)-(16) for equipment upgrades, constraints (18)-(19) for new equipment and constraint (21) for new storage tanks from part I. In equation (4), $Y_{t,s}$ summarizes the operational constraints (2)-(14) from part I for season t and scenario s, in which the variables for modes $y_{t,s}^m$ and transitions $y_{t,s}^{tr}$ are binary, and the variables for internal flowrates, inventories, sales and external product purchases, $y_{t,s}^c$, are continuous:

$$
Y_{t,s} = \left\{ y_{t,s} = \left(y_{t,s}^m, y_{t,s}^{tr}, y_{t,s}^c \right)^T, y_{t,s}^m \in \{0,1\}, y_{t,s}^{tr} \in \{0,1\}, y_{t,s}^c \geq 0 : B_{t,s} y_{t,s} \leq b_{t,s} \right\} \quad (6)
$$

While the index for hours h is omitted in (6), we would like to highlight that each set of operational constraints $Y_{t,s}$ represents a weekly scheduling problem with an hourly time discretization (168 hours). Note that the operational problems become independent of each other, once the investment decisions $x_{t'}$ are fixed since there is no inventory carry-over between adjacent

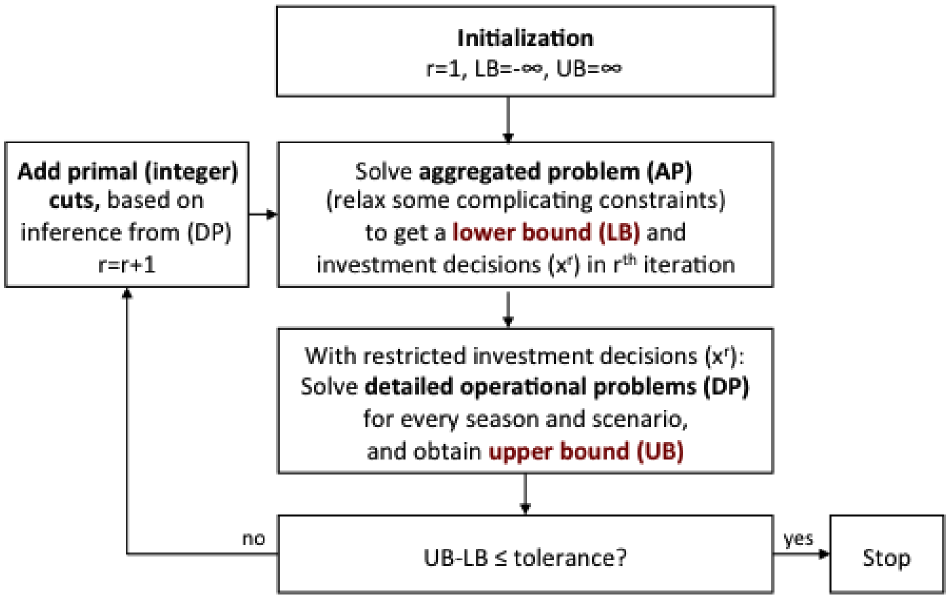

Figure 1: **Classical bi-level decomposition algorithm, in which (AP) is a relaxation of the original problem (P) and (DP) is a restriction of (P).**

seasons or scenarios, which we can exploit with our decomposition strategy in the following section.

3 Bi-level decomposition algorithm

The bi-level decomposition algorithm tackles the original problem (P) by alternately solving a *relaxation* and a *restriction* of (P). The *relaxation* of (P), denoted as aggregated problem (AP), is built with a subset of the original primal constraints, based on domain-specific knowledge. The idea is that not all primal information is needed in order to determine good values for the complicating variables of (P). Once (AP) is solved, the complicating variables are restricted by either fixing all of them to their respective values obtained from (AP), or only fixing the variables that were found to be zero in (AP). For the restricted complicating variables, the detailed problem (DP), which is a *restriction* of (P), is solved and *primal cuts* that are inferred from (DP) are added back to (AP). The algorithm iterates until the gap between the bounds obtained from (AP) and (DP) is within a predefined tolerance. The generic bi-level decomposition algorithm is shown in Fig. 1. It was proved by Iyer and Grossmann (1998) that the algorithm converges within a finite number of iterations for the given tolerance, or enumerates in the worst case all feasible combinations for the complicating variables.

The concept of complicating variables is the same as in Benders decomposition (Benders, 1962; Geoffrion, 1972). However, Benders decomposition tends to converge slowly if the underlying LP or NLP relaxation is weak (Sahinidis and Grossmann, 1991) since the dual cuts are weak, as well as essential problem information is missing in the master problem. The bi-level decomposition algorithm aims to address this issue by incorporating some or most of the original primal constraints in its “master problem” (AP).

3.1 Aggregated problem (AP)

It is important to recognize that there are multiple ways to construct (AP). We identify several constraints that seem to be less relevant in order to obtain a “good set” of investments for (P), of which we discuss two options for subsets of constraints in more detail.

The first option for a subset of constraints, which could be relaxed, contains the logic constraints (9)-(12) from part I, which are part of $Y_{t,s}$. The logic constraints (9)-(12) restrict the transitions between operating modes and include minimum up- and downtimes and transitional times. If we relax these logic constraints, the number of second-stage variables is also reduced since the transitional variables $y^{tr}_{t,s}$ are no longer present in the problem. While the number of constraints and variables is reduced significantly, this relaxation has the drawback that it still contains second-stage integer variables, and potentially underestimates operating expenditures greatly since all costs related to transitions are neglected.

The second option for a subset of constraints that could be relaxed, includes the integrality requirements for the second-stage variables for modes and transitions ($y^{m}_{t,s}$ and $y^{tr}_{t,s}$). While the problem size is not reduced, the problem becomes considerably easier to solve since only a small number of integer variables, namely the investment decisions $x_{t'}$, remain in the problem. Furthermore, transitional costs are not entirely neglected. However, if the formulation were not tight in terms of the second-stage integer variables, neither would be the relaxation.

We notice that two conflicting objectives need to be balanced: the tightness of the relaxation (AP) and its difficulty to solve, which are both problem-dependent. Our computational experience shows that the first option is faster to solve than the second option, despite the second-stage integer variables for operating modes. On the other hand, the relaxation of the second option is tighter since transitional cost are not entirely neglected.

However, an additional criterion could be to construct (AP) in a way that (AP) yields a convex relaxation once the complicating variables $x_{t'}$ are fixed, which facilitates the application of additional decomposition methods to (AP). From this perspective, the second option is more favorable. Interestingly, it corresponds to the convex relaxation suggested by Sundaramoorthy et al. (2012) for their capacity planning problem in the pharmaceutical industry within the non-convex generalized Benders decomposition scheme. While we pursue the second option, we note that the general framework of the bi-level decomposition algorithm provides additional tools to speed up the solution process, such as primal cuts that will be discussed later in this paper. Hence, (AP) is formulated in the following way:

$$
\begin{aligned}
(AP) \quad & \min \sum_{t' \in T_{invest}} c_{t'}^T x_{t'} + \sum_{t \in T, s \in S} \tau_{t,s} d_{t,s}^T y_{t,s} && (7)\\
& s.t. \sum_{t' \in T_{invest}} A_{0,t'} x_{t'} \leq b_0 && (8)\\
& \sum_{t' \in T_{invest}, t' \leq t} A_{1,t'} x_{t'} + B_1 y_{t,s} \leq b_1 \quad \forall t \in T, s \in S && (9)\\
& y_{t,s} \in \overline{Y}_{t,s} \quad \forall t \in T, s \in S && (10)\\
& x_{t'} \in \{0,1\}^n \quad \forall t' \in T_{invest} && (11)
\end{aligned}
$$

where $\overline{Y}_{t,s}$ is defined by relaxing the second-stage integrality constraints for $Y_{t,s}$:

$$
\overline{Y}_{t,s} = \left\{ y_{t,s} = \left(y_{t,s}^m, y_{t,s}^{tr}, y_{t,s}^c \right)^T, y_{t,s}^m, y_{t,s}^{tr}, y_{t,s}^c \geq 0 : B_{t,s} y_{t,s} \leq b_{t,s} \right\} \quad (12)
$$

Additional cuts that augment (AP) and facilitate the solution process will be described in section 3.3.

3.2 Detailed problem (DP)

Iyer and Grossmann (1998), as well as Erdirik-Dogan and Grossmann (2007), construct the detailed problem (DP) by augmenting the original problem constraints with constraints on the feasible region of the complicating variables. More specifically, only those complicating variables that correspond to zero in the solution of (AP) are fixed. Therefore, all subsets for a given set of investments are evaluated implicitly in (DP). However, in our case, the problem decouples in terms of seasons t and scenarios s only if *all* binary variables for investments, $x_{t'}$, are fixed, which is denoted as $x_{t'}^r$ (for the r^{th} iteration). Hence, we obtain the following definition for our detailed problem (DP^r):

$$
\begin{aligned}
(DP^r) \quad & \min \sum_{t' \in T_{invest}} c_{t'}^T x_{t'}^r + \sum_{t \in T, s \in S} \tau_{t,s} d_{t,s}^T y_{t,s} && (13)\\
& s.t. \sum_{t' \in T_{invest}, t' \leq t} A_{1,t'} x_{t'}^r + B_1 y_{t,s} \leq b_1 \quad \forall t \in T, s \in S && (14)\\
& y_{t,s} \in Y_{t,s} \quad \forall t \in T, s \in S && (15)
\end{aligned}
$$

In practice, we can solve (DP^r) individually for each season t and each scenario s.

3.3 Cuts for (AP)

In the original work on bi-level decomposition (Iyer and Grossmann, 1998; Erdirik-Dogan and Grossmann, 2007), three types of cuts have been proposed that can be added to the aggregated problem (AP) for each iteration of the bi-level decomposition algorithm: design cuts, superset cuts and subset cuts. The idea of design cuts is to formulate valid bounds for major continuous variables, e.g. levels for capacity expansions, that become active once the previously set of investments x^r evaluated in (DP) is selected again (based on a logic condition that is the same as the no-good cut by Balas and Jeroslow, 1972). Depending on the problem structure, superset cuts are formulated to exclude supersets of x^r. Analogously, subset cuts exclude any subset of x^r since (DP) is constructed in a way that all subsets of x^r are implicitly evaluated.

Unfortunately, we cannot directly apply any of these cuts. Superset and subset cuts are not applicable since we cannot predict the impact of an investment in (DP) based on the solution in (AP) due to the non-convexity of $Y_{t,s}$. Furthermore, related to how we construct (DP), i.e. fixing all investment variables to their respective values obtained from (AP) to exploit the decomposable structure, the subset cuts would not be valid either.

However, in the following we describe subset-type cuts for the second-stage value function that combine ideas from design and subset cuts, and do not cut off the optimal solution.

3.3.1 Subset-type cuts for second-stage value function

The idea of the subset-type cuts is to infer information on the (continuous) operating cost from (DP) for season t and scenario s rather than excluding subsets from the feasible space of (AP). The cuts are based on the following propositions:

Proposition 3.1 *Let $z^{*,r}_{DP,t,s}$ be the optimal objective value of the recourse function for season t and scenario s from (DP) for a given set of investments x^r. Then, any subset of investments of x^r will yield a second-stage recourse function value of at least $\tau_{t,s} d^T_{t,s} y_{t,s} \geq z^{*,r}_{DP,t,s}$.*

Proof All objective function cost coefficients are positive and the coefficients in $A_{1,t'}$ are such that the feasible region for $y_{t,s}$ is increased if any $x_{t'}$ is selected. Therefore, any subset of x^r will have a more restricted feasible region for $y_{t,s}$, compared to the feasible region for $y_{t,s}$ associated with x^r. Hence, $\tau_{t,s} d^T_{t,s} y_{t,s} \geq z^{*,r}_{DP,t,s}$ for any subset of x^r. ∎

Let us partition the set of investments N, in disjunctive investments D, which are investments that can be executed only once over the entire time horizon as by (2) (e.g. unique new equipment or equipment upgrades), and cumulative investments C, which are investments that can be executed multiple times (e.g. storage tanks), such that $N = D \cup C$ for any $t' \in T_{invest}$.

Proposition 3.2 *If a given set of investments x^r includes an element $l \in D$, then the timing of the investment $x^l_{t'}$ does not impact the bound on the second-stage value function for $\tau_{t,s} d^T_{t,s} y_{t,s}$ from proposition 3.1, as long $x^l_{t'}$ is executed for $t' \leq t$.*

Proof In the linking constraint (3), which corresponds to constraints (15)-(16) for equipment upgrades and constraints (18)-(19) for new equipment (part I), the increase of the feasible region for $y_{t,s}$ is independent of the exact timing of the investment $x_{t'}$, as long as $t' \leq t$. ∎

Proposition 3.3 *For a given set of investments x^r, the bound on $\tau_{t,s} d^T_{t,s} y_{t,s}$ from proposition 3.1 is invalid if an investment $l \in C$ is selected at least one time more than in iteration r for $t' \leq t$.*

Proof For a given set of investments x^r, the feasible region for $y_{t,s}$ increases further if an investment $l \in C$ is executed one more time for $t' \leq t$ due to the linking constraint (3) (e.g. cumulative increases for inventory bounds in constraint (21) from part I for storage tanks). Therefore, $\tau_{t,s} d^T_{t,s} y_{t,s} \leq z^{*,r}_{DP,t,s}$, which concludes the proof. ∎

Let us define the following sets, which partition the solution x^r for each iteration r according to whether investment l in time period t' in iteration r is a disjunctive or cumulative investment, and was selected or not.

$$LD^r_{0,t'} = \left\{ l : l \in D, x^l_{t'} = 0 \text{ in iteration } r \right\} \tag{16}$$

$$LD^r_{1,t'} = \left\{ l : l \in D, x^l_{t'} = 1 \text{ in iteration } r \right\} \tag{17}$$

$$LC^r_{0,t'} = \left\{ l : l \in C, x^l_{t'} = 0 \text{ in iteration } r \right\} \tag{18}$$

$$LC^r_{1,t'} = \left\{ l : l \in C, x^l_{t'} = 1 \text{ in iteration } r \right\} \tag{19}$$

Furthermore, we introduce as follows the set $\overline{LD}^r_{t,t'}$, which indicates whether a disjunctive investment has not been executed before season t:

$$\overline{LD}^r_{t,t'} = \left\{ l : l \in LD^r_{0,t'}, l \notin \bigcup_{\substack{t'' \in T_{invest} \\ t'' \leq t}} LD^r_{1,t''} \right\} \quad \forall t \in T, t' \in T_{invest}, t' \leq t, r \in R \tag{20}$$

The set $\overline{LC}_t^{l,r}$ indicates how often the cumulative investment $l \in C$ has been selected for $t' \leq t$ in iteration r:

$$\overline{LC}_t^{l',r} = \bigcup_{\substack{t' \in T_{invest} \\ t' \leq t}} LC_{1,t'}^{l',r} \quad \forall l' \in C, t \in T, r \in R \tag{21}$$

Based on propositions 3.1-3.3, we can formulate the following logic statement, for which, if it is true, the bound on the second-stage value function for season t and scenario s becomes invalid ($X_t^{l',r}$ is defined further below):

$$\bigvee_{\substack{l \in \overline{LD}_{t,t'}^r \\ t' \leq t \\ t' \in T_{invest}}} x_{t'}^l \bigvee_{l' \in C} X_t^{l',r} \qquad \forall t \in T, s \in S, r \in R \tag{22}$$

We can translate the logic statement (22) into the following cuts using propositional logic (Raman and Grossmann, 1993), assuming that the bound obtained from the previous iteration r is valid unless the logic statement becomes true:

$$\tau_{t,s} d_{t,s}^T y_{t,s} \geq z_{DP,t,s}^{*,r} \left(1 - \sum_{\substack{l \in \overline{LD}_{t,t'}^r \\ t' \leq t \\ t' \in T_{invest}}} x_{t'}^l - \sum_{l' \in C} X_t^{l',r} \right) \quad \forall t \in T, s \in S, r \in R \tag{23}$$

In equations (22)-(23), $X_t^{l',r}$ is defined as follows:

$$X_t^{l',r} = 1 \quad \Leftrightarrow \quad \text{Investment } l' \in C \text{ is selected more than } |\overline{LC}_t^{l',r}| \text{ times for } t' \leq t, t' \in T_{invest} \text{ in iteration } r \tag{24}$$

To establish this link between $x_{t'}^{l'}$ and $X_t^{l',r}$, we need to formulate the following two logic relationships:

$$\Rightarrow \ : \ X_t^{l',r} \cdot \left(|\overline{LC}_t^{l',r}| + 1 \right) \leq \sum_{\substack{t' \in T_{invest} \\ t' \leq t}} x_{t'}^{l'} \qquad \forall l' \in C, t \in T, r \in R \tag{25}$$

$$\Leftarrow \ : \ \sum_{\substack{t' \in T_{invest} \\ t' \leq t}} x_{t'}^{l'} \leq |\overline{LC}_t^{l',r}| + |T_{invest}| \cdot X_t^{l',r} \ \forall l' \in C, t \in T, r \in R \tag{26}$$

In summary, equations (23), (25) and (26) need to be included in (AP) to establish the subset-type cut based on previous (DP) solutions. As it will be shown in the results, these cuts can have a big impact in terms of a reduction of computational time.

3.3.2 No-good cut

Once a given set of investments x^r has been evaluated in (DP), the same set should not be visited again in (AP), which can be enforced by formulating the following no-good/integer cut (Balas and Jeroslow, 1972):

$$\sum_{\substack{t' \in T_{invest} \\ l \in L_1^r}} x_{t'}^l - \sum_{\substack{t' \in T_{invest} \\ l \in L_0^r}} x_{t'}^l - |L_1^r| + 1 \leq 0 \quad \forall r \in R \tag{27}$$

In (27), L_0^r and L_1^r are defined as follows (independent whether the investment is disjunctive or cumulative):

$$L_0^r = \left\{ l : x_{t'}^l = 0 \text{ in iteration } r \right\} \tag{28}$$

$$L_1^r = \left\{ l : x_{t'}^l = 1 \text{ in iteration } r \right\} \tag{29}$$

3.4 Discussion of scalability of the classical bi-level decomposition algorithm

While each operational subproblem (DP), (13)-(15), is an MILP that represents the detailed weekly schedule on an hourly basis, the solution process of (DP) scales well in the sense that it can potentially be parallelized since each operational problem for season t and scenario s can be solved individually. In contrast, the size of (AP), (7)-(11), and its associated solution time depends on the number of individual operational subproblems. For a large set of seasons and scenarios, (AP) will still be hard to solve. Therefore, we need additional algorithmic techniques to enable the solution of (AP) in a scalable manner.

4 Ingredients for an enhanced bi-level decomposition algorithm

In the following, we describe the ingredients for a decomposition algorithm that is based on the previously outlined bi-level scheme. First, we revisit the detailed problem (DP), and introduce a modification in the objective function that allows us to derive additional Lagrangean-type cuts that can be included in (AP) during the initialization phase to improve the tightness of (AP). Second, the application of Benders decomposition to (AP) is discussed, which facilitates the solution of (AP) independent of the number of seasons and scenarios.

4.1 Detailed problem (DP)

To enable the Lagrangean-type cuts, we reformulate the detailed problem (DP), which is defined by equations (13)-(15), by distributing the first-stage investment cost of (13) over all scenarios, in the spirit of the scenario decomposition approach (Caroe and Schulz, 1999). For this purpose we split the cost coefficients as follows: $c_{t'} = \sum_{t\in T} c_{t',t}$. The detailed problem can be written as:

$$
\begin{aligned}
(DP^r) \quad \min \quad & \sum_{t\in T, s\in S} \sum_{t'\in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t'}^r + \sum_{t\in T, s\in S} \tau_{t,s} d_{t,s}^T y_{t,s} && (30)\\
s.t. \quad & \sum_{t'\in T_{invest}, t'\leq t} A_{1,t'} x_{t'}^r + B_1 y_{t,s} \leq b_1 \quad \forall t\in T, s\in S && (31)\\
& y_{t,s} \in Y_{t,s} \quad \forall t\in T, s\in S && (32)
\end{aligned}
$$

4.1.1 Detailed subproblems for each season and scenario

In practice, we solve (DP) for each season t and scenario s independently, for which we define $(DP_{t,s}^r)$ as:

$$
\begin{aligned}
(DP_{t,s}^r) \quad \min \quad & \sum_{t'\in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t'}^r + \tau_{t,s} d_{t,s}^T y_{t,s} && (33)\\
s.t. \quad & \sum_{t'\in T_{invest}, t'\leq t} A_{1,t'} x_{t'}^r + B_1 y_{t,s} \leq b_1 && (34)\\
& y_{t,s} \in Y_{t,s} && (35)
\end{aligned}
$$

Let $z_{DP,t,s}^{*r}$ be the optimal solution of $(DP_{t,s}^r)$. Then, the optimal solution of the r^{th} iteration of (DP), z_{DP}^{*r}, is defined as $z_{DP}^{*r} = \sum_{t\in T, s\in S} z_{DP,t,s}^{*r}$.

4.1.2 Initial Lagrangean-type cuts

The solution of the non-convex problem (DP) is a time-consuming ingredient of our algorithm, and we would like to minimize the number of (DP) evaluations. As described previously, the reformulation of the objective function as per (30) allows us to solve a Lagrangean-type reformulation of (DP), denoted as (LG), during the initialization phase to find strong initial bounds. The idea is to duplicate the first-stage investment variables $x_{t'}$ for each season t and scenario s, denoted as $x_{t',t,s}$, and solve each subproblem individually such that the same investment decisions are not enforced across all scenarios. We obtain the formulation (LG), which is a non-convex relaxation of (DP) and is expected to yield good initial bounds as follows:

$$
\begin{aligned}
(LG)\quad \min\ & \sum_{t\in T, s\in S}\sum_{t'\in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t',t,s} + \sum_{t\in T, s\in S} \tau_{t,s} d_{t,s}^T y_{t,s} && && (36)\\
s.t.\ & \sum_{t'\in T_{invest}} A_{0,t'} x_{t',t,s} && \leq b_0 \quad \forall t\in T, s\in S && (37)\\
& \sum_{t'\in T_{invest}, t'\leq t} A_{1,t'} x_{t',t,s} + B_1 y_{t,s} && \leq b_1 \quad \forall t\in T, s\in S && (38)\\
& y_{t,s}\in Y_{t,s} && \forall t\in T, s\in S && (39)\\
& x_{t',t,s}\in\{0,1\}^n && \forall t'\in T_{invest}, t\in T, s\in S && (40)
\end{aligned}
$$

(LG) can be decomposed and formulated for each season t and scenario s as $(LG_{t,s})$:

$$
\begin{aligned}
(LG_{t,s})\quad \min\ & \sum_{t'\in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t',t,s} + \tau_{t,s} d_{t,s}^T y_{t,s} && (41)\\
s.t.\ & \sum_{t'\in T_{invest}} A_{0,t'} x_{t',t,s} \leq b_0 && (42)\\
& \sum_{t'\in T_{invest}, t'\leq t} A_{1,t'} x_{t',t,s} + B_1 y_{t,s} \leq b_1 && (43)\\
& y_{t,s}\in Y_{t,s} && (44)\\
& x_{t',t,s}\in\{0,1\}^n \quad \forall t'\in T_{invest} && (45)
\end{aligned}
$$

Let $z_{LG,t,s}^* = \sum_{t'\in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t',t,s}^* + \tau_{t,s} d_{t,s}^T y_{t,s}^*$ be the solution of $(LG_{t,s})$. Note that $z_{LG,t,s}^*$ yields a valid lower bound for the cost in season t and scenarios s in $(DP_{t,s})$:

$$
\sum_{t'\in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t'} + \tau_{t,s} d_{t,s}^T y_{t,s} \geq z_{LG,t,s}^* \quad \forall t\in T, s\in S \qquad (46)
$$

Remark: From a Lagrangean relaxation perspective, we relax the non-anticipativity constraints for $x_{t',t,s}$, which enforce equality of $x_{t',t,s}$ across all scenarios, and solve the Lagrangean relaxation with the Lagrangean multipliers set to zero once during the initialization phase.

4.2 Decomposition of (AP)

The bottleneck of the classical bi-level decomposition is the aggregated problem (AP), which grows as the number of seasons and scenarios increases.

Interestingly, (AP) has only a few binary variables, the first-stage investment decisions, which complicate the solution of the problem. If we fix the first-stage investment decisions, (AP) decouples into individual subproblems, which only have continuous variables and are linear. For this setup, the application of Benders decomposition to (AP) is a natural one that we discuss in the following.

Note that (AP) could also be decomposed with Lagrangean decomposition, which would lead to a hybrid scheme similar to the ones described by Calfa et al. (2013) and Terrazas-Moreno and Grossmann (2011). However, both report dual gaps. Furthermore, Terrazas-Moreno and Grossmann (2011) remove the upper bounding procedure of (AP), which potentially weakens the strength of the (AP) bound.

In general, the application of Lagrangean decomposition can lead to the presence of a duality gap, which requires further branch and bound search. Furthermore, the update of the Lagrangean multipliers, as well as the application of a heuristic to find first-stage feasible solutions, are known to be difficult for Lagrangean decomposition. Additionally, the addition of integer cuts like the ones described in section 3.3 to the Lagrangean dual requires the application of a somewhat complicated re-optimization scheme (Frangioni, 2005) if cutting planes are used to update the Lagrangean multipliers.

4.2.1 Benders subproblems for (AP)

We apply Benders decomposition for (AP), which decouples into subproblems for each season t and scenario s. Note that we need to modify the objective function according to the change in (DP)'s objective function (33). We formulate the Benders subproblem in its primal form as follows for a given set of investments $x_{t'}^k$, where $k \in K$ is the counter of the inner loop for the decomposition of (AP):

$$(APp_{t,s}^k) \quad \min \quad \sum_{t' \in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t'}^k + \tau_{t,s} d_{t,s}^T y_{t,s} \tag{47}$$

$$s.t. \quad B_1 y_{t,s} \leq b_1 - \sum_{\substack{t' \in T_{invest} \\ t' \leq t}} A_{1,t'} x_{t'}^k \tag{48}$$

$$y_{t,s} \in \overline{Y}_{t,s} \tag{49}$$

Let $u_{t,s}$ be the dual multiplier associated with constraint (48) and $v_{t,s}$ be the dual multiplier of constraint (49), which represents $B_{t,s} y_{t,s} \leq b_{t,s}$ according to the definition of $\overline{Y}_{t,s}$ in (12). Then, the dual of $(APp_{t,s})$ can be written as:

$$
\begin{aligned}
(APd^k_{t,s}) \quad \max \quad & \sum_{t' \in T_{invest}} \tau_{t,s} c^T_{t',t} x^k_{t'} && (50)\\
& + \left(\sum_{\substack{t' \in T_{invest}\\ t' \le t}} A_{1,t'} x^k_{t'} - b_1 \right)^T u_{t,s} - b^T_{t,s} v_{t,s} \\
s.t. \quad & -u^T_{t,s} B_1 - v^T_{t,s} B_{t,s} \le \tau_{t,s} d_{t,s} && (51)\\
& u_{t,s}, v_{t,s} \ge 0 && (52)
\end{aligned}
$$

Let $y^{*k}_{t,s}$ be the solution of $(APp^k_{t,s})$ and $(u^{*k}_{t,s}, v^{*k}_{t,s})$ be the solution of $(APd^k_{t,s})$. Let us define $z^{*k}_{AP,t,s}$, in the following way (equivalence due to strong duality):

$$
\begin{aligned}
z^{*k}_{AP,t,s} &= \sum_{t' \in T_{invest}} \tau_{t,s} c^T_{t',t} x^k_{t'} + \tau_{t,s} d^T_{t,s} y^{*k}_{t,s} \\
&= \sum_{t' \in T_{invest}} \tau_{t,s} c^T_{t',t} x^k_{t'} + \left(\sum_{\substack{t' \in T_{invest}\\ t' \le t}} A_{1,t'} x^k_{t'} - b_1 \right)^T u^{*k}_{t,s} - b^T_{t,s} v^{*k}_{t,s}
\end{aligned}
$$

The optimal solution of the k^{th} iteration of (AP), z^{*k}_{AP}, is therefore defined as $z^{*k}_{AP} = \sum_{t \in T, s \in S} z^{*k}_{AP,t,s}$.

4.2.2 Dual Benders optimality cuts

With the dual solution $(u^{*k}_{t,s}, v^{*k}_{t,s})$ from $(APp^k_{t,s})$, we can derive Benders optimality cuts for (AP). The Benders cuts are based on weak duality since $(u^k_{t,s}, v^k_{t,s})$ yields a lower bound even for a $x_{t'}$ different from $x^k_{t'}$. The classical Benders cut (Benders, 1962) can be written as:

$$
\eta_{MP} \ge \sum_{\substack{t \in T\\ s \in S}} \sum_{t' \in T_{invest}} \tau_{t,s} c^T_{t',t} x_{t'} + \left(\sum_{\substack{t' \in T_{invest}\\ t' \le t}} A_{1,t'} x_{t'} - b_1 \right)^T u^{*k}_{t,s} - b^T_{t,s} v^{*k}_{t,s} \quad \forall k \in K \tag{53}
$$

Alternatively, the optimality cut can be written as multi-cut version for each subproblem (Birge and Louveaux, 1988), where $\theta_{t,s}$ is an approximation of the second-stage value function for season t and scenario s:

$$\eta_{MP} \geq \sum_{\substack{t \in T \\ s \in S}} \theta_{t,s} \tag{54}$$

$$\theta_{t,s} \geq \sum_{t' \in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t'} + \left(\sum_{\substack{t' \in T_{invest} \\ t' \leq t}} A_{1,t'} x_{t'} - b_1 \right)^T u_{t,s}^{*k} - b_{t,s}^T v_{t,s}^{*k} \tag{55}$$

$$\forall t \in T, s \in S, k \in K$$

While the multi-cut version (54)-(55) provides faster convergence due to increased strength of the dual information, it increases the problem size compared to the single-cut version (53).

Note that due to the introduction of external product purchases into the operational subproblems, feasibility is always guaranteed and we have complete recourse. Therefore, we do not include Benders feasibility cuts.

4.2.3 No-good cut for (AP)

Since we decomposed (AP), which acts as a solution generator for (DP), we can move the no-good cut (27) from the outer loop to the inner loop. Therefore, we re-define the corresponding sets L_0^k and L_1^k over the inner iteration counter $k \in K$:

$$L_0^k = \left\{ l : x_{t'}^l = 0 \text{ in iteration } k \right\} \tag{56}$$

$$L_1^k = \left\{ l : x_{t'}^l = 1 \text{ in iteration } k \right\} \tag{57}$$

The no-good cut is than written as:

$$\sum_{\substack{t' \in T_{invest} \\ l \in L_1^k}} x_{t'}^l - \sum_{\substack{t' \in T_{invest} \\ l \in L_0^k}} x_{t'}^l - |L_1^k| + 1 \leq 0 \quad \forall k \in K \tag{58}$$

4.2.4 Formulation of the Master Problem

The master problem serves the dual purpose of aggregating cuts from the decomposition of (AP), as well as primal cuts from the bi-level scheme, since it is also a relaxation of (DP). We include the dual Benders multi-cuts (54)-(55), the no-good cut (58) and the first-stage constraints (2) in the master problem. The tightness of the master problem is improved by adding the initial Lagrangean-type cuts (46) and the subset-type cut (23), modified for the change in (DP)'s objective function, with additional logic constraints (25)-(26). Hence, the master problem (MP^{k+1}) is defined as:

$$
\begin{aligned}
& (MP^{k+1}) \\
\min \quad & \eta_{MP} && (59)\\
s.t. \quad & \eta_{MP} \geq \sum_{\substack{t\in T\\ s\in S}} \theta_{t,s} && (60)\\
& \theta_{t,s} \geq \sum_{t'\in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t'} + \left(\sum_{\substack{t'\in T_{invest}\\ t'\leq t}} A_{1,t'} x_{t'} - b_1 \right)^T u_{t,s}^{*k} - b_{t,s}^T v_{t,s}^{*k} && (61)\\
& \qquad \forall t\in T, s\in S, k\in K \\
& \theta_{t,s} \geq z_{LG,t,s}^{*} \qquad \forall t\in T, s\in S && (62)\\
& \theta_{t,s} \geq \sum_{t'\in T_{invest}} \tau_{t,s} c_{t',t}^T x_{t'} + z_{DP,t,s}^{*,r} \left(1 - \sum_{\substack{l\in \overline{LD}_{t,t'}^{r}\\ t'\leq t\\ t'\in T_{invest}}} x_{t'}^{l} - \sum_{l'\in C} X_t^{l',r} \right) && (63)\\
& \qquad \forall t\in T, s\in S, r\in R \\
& X_t^{l',r} \cdot \left(|\overline{LC}_t^{l',r}| + 1 \right) \leq \sum_{\substack{t'\in T_{invest}\\ t'\leq t}} x_{t'}^{l'} \qquad \forall l'\in C, t\in T, r\in R && (64)\\
& \sum_{\substack{t'\in T_{invest}\\ t'\leq t}} x_{t'}^{l'} \leq |\overline{LC}_t^{l',r}| + |T_{invest}| \cdot X_t^{l',r} \qquad \forall l'\in C, t\in T, r\in R && (65)\\
& \sum_{l\in L_1^k} x_{t'}^{l} - \sum_{l\in L_0^k} x_{t'}^{l} - |L_1^k| + 1 \leq 0 \qquad \forall k\in K && (66)\\
& \sum_{t'\in T_{invest}} A_{0,t'} x_{t'} \leq b_0 && (67)\\
& x_{t'} \in \{0,1\}^n \qquad \forall t'\in T_{invest} && (68)\\
& X_t^{l',r} \in \{0,1\} \qquad \forall t\in T, l'\in C, r\in R && (69)\\
& \eta_{MP} \in \mathbb{R}^1,\ \theta_s \in \mathbb{R}^{|S|} && (70)
\end{aligned}
$$

The master problem (MP^{k+1}) yields x^{k+1}, which is evaluated in $(APp_{t,s}^{k+1})$, and the optimal objective function value z_{MP}^{*k+1} is a valid lower bound for (P).

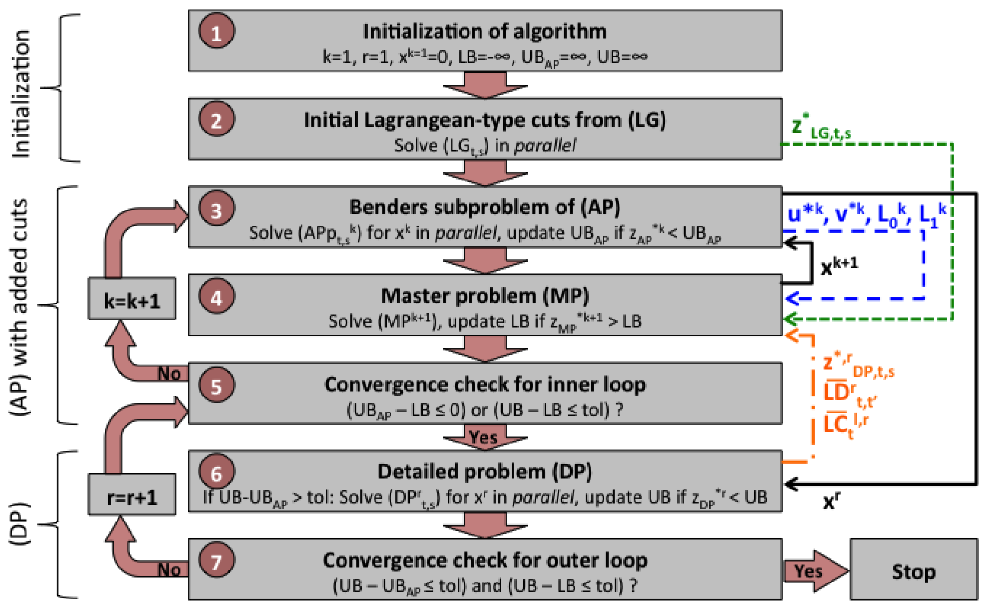

Figure 2: **Algorithmic flow (bold arrows) and flow of information (thin arrows) for our enhanced hybrid bi-level decomposition scheme.**

5 The enhanced hybrid bi-level decomposition algorithm

5.1 Algorithm EBL-multi+LG+subset

In the following we describe the enhanced hybrid bi-level decomposition algorithm, which is depicted in Fig. 2.

1. **Initalization**
   Set $k = 1$, $K = \{\}$, $K' = \{\}$, $r = 1$, $R = \{\}$, $x^{k=1} = 0$,
   $LB = -\infty$, $UB_{AP} = \infty$, $UB = \infty$, $tol \geq 0$,
   Go to step 2

2. **Initial Lagrangean-type cuts from (LG)**
   Solve $(LG_{t,s})$, (41)-(45), $\forall t \in T, s \in S$ in parallel and store $z^*_{LG,t,s}$
   Go to step 3

3. **Benders subproblems of (AP)**
   For given x^k, solve $(APp^k_{t,s})$, (47)-(49), $\forall t \in T, s \in S$ in parallel
   Store dual multipliers u^{*k}, v^{*k} and generate sets L^k_0, L^k_1, $K = K \cup \{k\}$
   If $z^{*k}_{AP} < UB_{AP}$: $UB_{AP} = z^{*k}_{AP}$ and $x^r = x^k$, $k' = k$
   Go to step 4

4. **Master problem (MP)**
   For given u^{*k}, v^{*k}, L_0^k, L_1^k, $z^*_{LG,t,s}$, $z^{*,r}_{DP,t,s}$, $\overline{LD}^r_{t,t'}$, $\overline{LC}^{l,r}_t$:
   Solve (MP^{k+1}), (59)-(70), and store solution as x^{k+1}
   If $z^{*k+1}_{MP} > LB$: $LB = z^{*k+1}_{MP}$
   Go to step 5

5. **Convergence check for inner loop**
   If $UB_{AP} - LB \leq 0$ or $UB - LB \leq tol$: Go to step 6
   Else: Set $k = k + 1$, go to step 3

6. **Detailed problem (DP)**
   If $UB - UB_{AP} > tol$:

   For given x^r, solve $(DP^r_{t,s})$, (33)-(35), $\forall t \in T, s \in S$ in parallel

   $R = R \cup \{r\}$, $K' = K' \cup \{k'\}$

   If $z^{*k}_{DP} < UB$: $UB = z^{*k}_{DP}$, solution $= (x^r, y^*_{t,s})_{DP}$

   If $K \setminus K' = \emptyset$: $UB_{AP} = \infty$

   Else: $k' = \text{argmin}_{k \in K \setminus K'} \{z^{*k}_{AP}\}$, $UB_{AP} = z^{*k'}_{AP}$, $x^r = x^{k'}$
   Go to step 7

7. **Convergence check for outer loop**
   If $UB - UB_{AP} \leq tol$ and $UB - LB \leq tol$: Stop and return solution
   Else: Set $r = r + 1$, go to step 5

5.2 Further variants of the algorithm

In this section, we define "reduced" variants of the previously described algorithm *EBL-multi+LG+subset* in order to test the impact of the individual components of the algorithm.

5.2.1 EBL-single

In the first variant, we remove the Lagrangean-type cuts (step 2) completely, and do not use the subset type cuts, equations (63)-(65), in the master problem. Furthermore, we use the classical Benders cut, equation (53), in place of the multi-cut (60)-(61).

Remark: The bi-level decomposition algorithm with classical Benders single cuts for (AP), denoted by EBL-single, is equivalent to non-convex generalized Benders decomposition applied to two-stage stochastic programming problem, as described by Sundaramoorthy et al. (2012) for a capacity planning problem in the pharmaceutical industry.

5.2.2 EBL-multi

In this variant, we remove the Lagrangean-type cuts (step 2) completely, and do not use the subset type cuts, equations (63)-(65), in the master problem. The rest of the algorithm is as described in section 5.1.

5.2.3 EBL-multi+LG

The only difference between EBL-multi+LG and the algorithm we described in section 5.1 is that we do not use the subset-type cuts, equations (63)-(65), in the master problem.

5.3 Convergence of the algorithm

A formal proof of convergence for the bi-level decomposition algorithm was given by Iyer and Grossmann (1998), and a proof for the non-convex generalized Benders decomposition was stated by Barton and co-workers (2011, 2012). Therefore, we omit the proof in this paper.

Furthermore, we would like to comment on the issue of the tightness of the relaxation (AP). If the relaxation of (AP) is weak, we might enumerate all investments and observe a slow progress in the lower bound obtained from the master problem. The addition of the Lagrangean-type cuts from (LG) and the subset-type cuts is motivated by this potential issue.

6 Computational aspects

6.1 Parallel Implementation

The Lagrangean-type relaxation $(LG_{t,s})$, the Benders subproblems for $(APp^k_{t,s})$ and the detailed problem $(DP^r_{t,s})$ can be decomposed into operational subproblems for each season t and scenario s. Note that these subproblems can be solved in parallel. Hence, we use the GAMS Grid Computing environment (Bussieck, Ferris and Meeraus, 2009) to parallelize the solution process for $(LG_{t,s})$, $(APp^k_{t,s})$ and $(DP^r_{t,s})$. We run our implementation on a machine with 8 cores, which were utilized in the parallelization. The same code could be executed on a 'real' grid computing environment, in which we envision that a much larger number of scenarios could be easily solved in parallel.

6.2 Discussion of case study setup

We solve the same test cases (case 1-4 for varying baseline demand, growth rate, demand distribution and prices for external product purchases) that were reported in part I of the paper on the same computing machine, an Intel i7-2600 (3.40 GHz) machine with 8 processors and 8 GB RAM. The commercial solver CPLEX 12.4.0.1 was employed in GAMS 23.9.1 (Brooke et al., 2012). The termination criterion was set to 0.5% optimality gap. Note that the algorithm could accept and converge for smaller values as well, but we choose 0.5% to make our numbers comparable with the full-space approach, in which we allowed the same final gap.

As we can see in Table 1, the resulting full-space model is very large, as it has almost 1 million constraints and more than 2 million variables, of which there are more than 200,000 binaries. In contrast, we can observe that the problem sizes for each subproblem for one season and scenario in $(DP^r_{t,s})$, $(LG_{t,s})$ and $(AP^k_{t,s})$ are (not surprisingly) much smaller.

Table 1: **Sizes of the resulting optimization problems in terms of constraints and variables. Except for (MP), which depends on the number of iterations, the sizes are the same for all problem instances (1-4). Note that the numbers for (MP) correspond to case 3 for the EBL-multi+LG+subset scheme.**

	Full-space	$(DP^r_{t,s})$	$(LG_{t,s})$	$(AP^k_{t,s})$	(MP) k=1, r=1	(MP) k=14, r=6
Number of op. problems	60	1	1	1	-	-
Constraints	915,270	14,958	14,958	14,958	184	1,877
Variables	2,388,984	39,840	39,840	39,840	141	441
Binary Var.	221,780	3,696	3,716	0	20	320
Binary Invest.	20	0	20	0	20	20

We compare the full-space solution (as reported in part I of the paper for which we used the parallel computing capabilities of CPLEX by setting `threads=8`) with four different variants of the decomposition algorithm, and investigate the impact of each individual component in the decomposition algorithm. The four variants are described in detail in section 5, and we shortly restate them here:

1. **EBL-single** Hybrid bi-level decomposition with single cut Benders decomposition for (AP)

2. **EBL-multi** Hybrid bi-level decomposition with multi-cut Benders decomposition for (AP)

3. **EBL-multi+LG** Hybrid bi-level decomposition with multi-cut Benders decomposition for (AP) and the initial Lagrangean-type cuts from (LG)

4. **EBL-multi+LG+subset** Hybrid bi-level decomposition with multi-cut Benders decomposition for (AP), the initial Lagrangean-type cuts

from (LG) and the subset-type cuts from (DP), i.e. like the algorithm we described in section 5.

6.3 Discussion of results

Table 2: **Comparison of computational times for the full-space method and the different variants of the hybrid bi-level decomposition algorithm.** Allowed gap in full-space: 0.50%. [a]: out of memory; [b]: terminated after 80 hours of computation

method	property	case 1	case 2	case 3	case 4
full-space	time (s)	**78810**	**67137**	**288000**	**154312**
full-space	gap	1.32%[a]	0.50%	3.23% [b]	0.60% [a]
All EBL-schemes	gap	<0.50%	<0.50%	<0.50%	<0.50%
EBL-single	time (s)	721	4321	9832	16345
EBL-single	AP	14	28	43	33
EBL-single	DP	1	5	16	14
EBL-multi	time (s)	583	3820	9097	16009
EBL-multi	AP	11	18	28	24
EBL-multi	DP	1	5	16	14
EBL-multi+LG	time (s)	403	3930	6415	11925
EBL-multi+LG	AP	1	13	21	17
EBL-multi+LG	DP	1	5	14	11
EBL-multi+LG+subset	time (s)	**403**	**1869**	**2846**	**2436**
EBL-multi+LG+subset	AP	1	9	14	9
EBL-multi+LG+subset	DP	1	2	7	3

Due to confidentiality issues, we cannot disclose the final objective function values. However, we can confirm that all hybrid bi-level decomposition schemes converge to the same final solution for each case. Furthermore, the same investments are proposed in the full-space method and in each of the hybrid bi-level decomposition schemes. Thus, we can conclude that the full-space method's major issue is to prove optimality due to the large number of binary variables.

We report the total execution times in seconds, and the number of (AP) and (DP) evaluations in Table 2. The most time-consuming part of the algorithm is the detailed problem (DP), which is a detailed scheduling problem (MILP) for an entire week on an hourly for each season t and scenario s. In fact, (DP) can consume 90% or more of the total execution time since one (AP) evaluation takes only 40-50s. Therefore, the impact on execution time of the multi cuts in *EBL-multi* is small compared to *EBL-single* since it only reduces the number of (AP) evaluations. In contrast, the Lagrangean-type cuts from (LG) reduce the number of (DP) iterations in the difficult cases 3 and 4, and saves 20-25% in computational time. The addition of the subset-type cuts (*EBL-multi+LG+subset*) has the largest impact on computational time since we can observe another 50-80% reduction compared to *EBL-multi+LG* and a speed-up of 45-85% compared to the *EBL-single* scheme. If we compare the times for our most advanced scheme (*EBL-multi+LG+subset*) with the full space method we can observe a speed-up of up to two orders of magnitude across all four test cases.

7 Conclusion

In this paper (part II), we have described an enhanced hybrid bi-level decomposition scheme that combines bi-level decomposition with Benders decomposition and additional cuts to strengthen the relaxation. We have solved a set of instances for a large multi-scale capacity planning problem based on industrial data with the decomposition scheme, and we were able to reduce the computational time by up to two orders of magnitude compared to the full-space method and by 45-85% compared to a pure non-convex generalized Benders approach. The results were obtained due to additional cuts obtained from a Lagrangean-type relaxation and subset-type cuts from the detailed problem (DP) that exploit the linking structure between investment and operational variables in the problem. For future work, it would be interesting to explore other cuts from the structure of (DP) or other (non-convex) relaxations of (DP) that can be utilized to yield strong bounds in order to further reduce the number of (DP) evaluations.

Acknowledgments

We would like to thank Praxair, Inc., and the National Science Foundation for financial support under grant #1159443.

References

[1] Balas, E.; Jeroslow, R.G. Canonical Cuts on the Unit Hypercube. *SIAM Journal of Applied Mathematics*, 23:61–69, 1972.

[2] Benders, J.F. Partitioning Procedures for Solving Mixed-variables Programming Problems. *Numerische Mathematik*, 4:238–252, 1962.

[3] Birge, J.R.; Louveaux, F. *Introduction to Stochastic Programming.* Springer Series in Operations Research. Springer, 2011.

[4] Birge, J.R.; Louveaux, F.V. A Multicut Algorithm for Two-stage Stochastic Linear Programs. *European Journal of Operational Research*, 34:384–392, 1988.

[5] Brooke, A.; Kendrick, D.; Meeraus, A. GAMS: A Users Guide, Release 23.9.1. *The Scientific Press, South San Francisco*, 2012.

[6] Bussieck, M.R.; Ferris, M.C.; Meeraus, A. Grid-Enabled Optimization with GAMS. *INFORMS Journal on Computing*, 21:349–362, 2009.

[7] Calfa, B. A.; Agarwal, A.; Grossmann, I. E.; Wassick, J. M. Hybrid Bilevel-Lagrangean Decomposition Scheme for the Integration of Planning and Scheduling of a Network of Batch Plants. *Industrial & Engineering Chemistry Research*, 52:2152–2167, 2013.

[8] Caroe, C. C.; Schultz, R. Dual Decomposition in Stochastic Integer Programming. *Operations Research Letters*, 24:37–45, 1999.

[9] Erdirik-Dogan, M.; Grossmann, I.E. Simultaneous planning and scheduling of single-stage multi-product continuous plants with parallel lines. *Computers & Chemical Engineering*, 32:2664–2683, 2008.

[10] Frangioni, A. About Langrangian Methods in Integer Optimization. *Annals of Operations Research*, 139:163–193, 2005.

[11] Geoffrion, A.M. Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.

[12] Guignard, M. Lagrangean Relaxation. *Top*, 11:151–228, 2003.

[13] Guignard, M.; Kim, S. Lagrangean Decomposition - a Model Yielding Stronger Lagrangean Bounds. *Mathematical Programming*, 39:215–228, 1987.

[14] Iyer, R.; Grossmann, I.E. A Bilevel Decomposition Algorithm for Long-range Planning of Process Networks. *Industrial & Engineering Chemistry Research*, 37:474, 1998.

[15] Karuppiah, R.; Grossmann, I.E. A Lagrangean based Branch-and-cut Algorithm for Global Optimization of Nonconvex Mixed-integer Nonlinear Programs with Decomposable Structures. *Journal of Global Optimization*, 41:163–186, 2008.

[16] Li, X.; Tomasgard, A.; Barton, P.I. Nonconvex Generalized Benders Decomposition for Stochastic Separable Mixed-Integer Nonlinear Programs. *Journal of Optimization Theory and Applications*, 151:425–454, 2011.

[17] Raman, R.; Grossmann, I.E. Modeling and Computational Techniques for Logic Based Integer Programming. *Computers & Chemical Engineering*, 18:563, 1993.

[18] Sundaramoorthy, A.; Li, X.; Evans, J. M. B.; Barton, P.I. Capacity Planning under Clinical Trials Uncertainty in Continuous Pharmaceutical Manufacturing, 2: Solution Method. *Industrial & Engineering Chemistry Research*, 51:13703–13711, 2012.

[19] Terrazas-Moreno, S.; Grossmann, I.E. A Multiscale Decomposition Method for the Optimal Planning and Scheduling of Multi-site Continuous Multiproduct Plants. *Chemical Engineering Science*, 66:4307–4318, 2011.

[20] van den Heever, S. A.; Grossmann, I. E. Disjunctive Multiperiod Optimization Methods for Design and Planning of Chemical Process Systems. *Computers & Chemical Engineering*, 23:1075, 1999.

[21] Van Slyke, R.; Wets, R.J.B. L-shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. *SIAM Journal on Applied Mathematics*, 17:638–663, 1969.

[22] You, F.; Grossmann, I. E.; Wassick, J. M. Multisite Capacity, Production and Distribution Planning with Reactor Modifications: MILP Model, Bilevel Decomposition Algorithm vs. Lagrangean Decomposition Scheme. *Industrial & Engineering Chemistry Research*, 50:4831–4849, 2011.