

10-24-2013

Data-driven multi-stage scenario tree generation via statistical property and distribution matching

Bruno A. Calfa
Carnegie Mellon University

Anshul Agarwal
Dow Chemical

Ignacio E. Grossmann
Carnegie Mellon University, grossmann@cmu.edu

John Wassick
Dow Chemical

Follow this and additional works at: <http://repository.cmu.edu/cheme>

 Part of the [Chemical Engineering Commons](#)

Published In

Computers and Chemical Engineering, 68, 7-23.

This Article is brought to you for free and open access by the Carnegie Institute of Technology at Research Showcase @ CMU. It has been accepted for inclusion in Department of Chemical Engineering by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

Data-Driven Multi-Stage Scenario Tree Generation via Statistical Property and Distribution Matching

Bruno A. Calfa*, Anshul Agarwal†, Ignacio E. Grossmann*, John M. Wassick†

October 24, 2013

Abstract

The objective of this paper is to bring systematic methods for scenario tree generation to the attention of the Process Systems Engineering community. In this paper, we focus on a general, data-driven optimization-based method for generating scenario trees, which does not require strict assumptions on the probability distributions of the uncertain parameters. This method is based on the Moment Matching Problem (MMP), originally proposed by [Høyland & Wallace \(2001\)](#). In addition to matching moments, and in order to cope with potentially under-specified MMP, we propose matching (Empirical) Cumulative Distribution Function information of the uncertain parameters. The new method gives rise to a Distribution Matching Problem (DMP) that is aided by predictive analytics. We present two approaches for generating multi-stage scenario trees by considering time series modeling and forecasting. The aforementioned techniques are illustrated with a motivating production planning problem with uncertainty in production yield and correlated product demands.

Keywords: Process Systems Engineering, Stochastic Programming, Scenario Generation, Distribution Matching Problem, Time Series Forecasting, Analytics

1 Introduction

The importance of accounting for uncertainty in mathematical optimization was recognized in its early days in the seminal and influential paper by George B. Dantzig ([Dantzig, 1955](#)). Two of the current popular optimization frameworks that incorporate uncertainty in the modeling stage are Robust Optimization ([Ben-Tal, Ghaoui, & Nemirovski, 2009](#)) and Stochastic Programming ([Birge & Louveaux, 2011](#)). In this paper, we focus on Stochastic Programming (SP) and address the issue of scenario generation.

To illustrate the many possible sources of uncertainty in Process Systems Engineering (PSE), consider as an example a production planning problem for a network of chemical plants. Planning decisions usually span multiple time periods and generally involve, but are

*Department of Chemical Engineering. Carnegie Mellon University. Pittsburgh, PA, 15213, USA.

†The Dow Chemical Company. Midland, MI, 48674, USA.

not limited to determining the amount of raw materials to be purchased by each plant, the production and inventory levels at each plant, the transportation of intermediate and finished products between different locations, and meeting the forecast demand. It is clear that all those decisions may be subject to some kind of uncertainty. For instance, the availability of a key raw material may be uncertain, *i.e.* there may be shortage for certain months in a year. Another example is the possibility of mechanical failure of pieces of equipment in a plant or its complete unplanned shutdown, which affects the entire network. Two types of uncertainty are reported in the literature (Goel & Grossmann, 2006): exogenous (*e.g.*, market) and endogenous (*e.g.*, decision-dependent). A review on optimization methods with exogenous uncertainties can be found in Sahinidis (2004).

A central aspect of Stochastic Programming is the definition of *scenarios*, which describe possible values that the uncertain parameters or stochastic processes may take. Applications in PSE that make explicit use of scenarios expand multiple areas and time scales. Some representative examples are: dynamic optimization (Abel & Marquardt, 2000), scheduling (Guilén, Espuña, & Puigjaner, 2006; Colvin & Maravelias, 2009; Pinto-Varela, Barbosa-Povoa, & Novais, 2009), planning (Sundaramoorthy, Evans, & Barton, 2012; Li & Ierapetritou, 2011; You, Wassick, & Grossmann, 2009; Gupta & Grossmann, 2012), and synthesis and design (Kim, Realff, & Lee, 2011; Chen, Adams II, & Barton, 2011). The most common assumption made in the works listed before is that the scenario tree is *given* (probabilities and values of uncertain parameters at every node are known). That is, the “true” probability distributions are known, and the uncertainty typically is characterized by arbitrary deviations from some average value based on minimum and maximum values (for instance: low, medium, and high values with probabilities arbitrarily chosen).

Researchers have also developed decomposition algorithms to tackle large-scale and real-world instances that originate from explicitly considering scenarios in optimization problems. We argue that it is equally important to generate scenario trees that satisfactorily capture the uncertainty in a given problem, as the quality of the solution to the SP problem is directly influenced by the accuracy of the scenarios. Therefore, it is important to apply systematic scenario generation methods instead of making assumptions that may be questionable. King & Wallace (2012) wrote an excellent book on the challenges of optimization modeling under uncertainty. The authors also discuss the importance of generating meaningful scenarios (see Chapter 4), as modeling with SP results in a framework with practical and robust decision-making capability.

These data-driven approaches to optimization problems have become common in the Operations Research and Management Science communities, and are an example of what is called *Business Analytics* (BA) (Bartlett, 2013). After the data collection and management phase, BA leverages data analysis to make analytics-based decisions that can be divided into three general layers: descriptive (querying and reporting, databases), predictive (forecasting and simulation), and prescriptive (deterministic and stochastic optimization) (Davenport & Harris, 2007). The data-driven scenario generation method described in this paper can be linked with the descriptive and predictive layers, and then used for decision-making in the prescriptive layer.

It is worth noting that, even though not usually regarded as a scenario generation method, the Sample Average Approximation (SAA) method (Kleywegt, Shapiro, & Homem-de-Mello, 2001; Shapiro, 2006) can be used to approximate the continuous probability distribution

assumed for the uncertain parameters. Specifically, the distributions are sampled, for instance via Monte Carlo sampling, and the expected value function is approximated by the corresponding sample average function, which is repeatedly solved until some convergence criterion is met. The size of the sample must be such that a degree of confidence on the final objective function value is satisfied. In addition, the sampling step becomes more complicated in Multi-Stage SP (MSSP), as *conditional* sampling is required for the SAA method to produce *consistent* estimators. Conditioning on previous events also plays a key role in the moment matching method as discussed later in the paper.

The goal of this paper is to bring systematic methods for scenario tree generation to the attention of the Process Systems Engineering community, and give an organizational structure to the formulations proposed in the literature thus far. We describe in detail the moment matching method for scenario tree generation. Different formulations of the MMP are presented. The main inputs or parameters to the MMP are the statistical moments of either time-independent random variables or stochastic processes. For the latter, statistical properties can be obtained through the aid of time series forecasting models as will be demonstrated. In order to cope with under-specified MMPs, we propose an extension to the MMP, called Distribution Matching Problem (DMP), in which cumulative distribution data are also matched. For completeness, we briefly present the ideas of scenario reduction (Dupačová, Gröwe-Kuska, & Römisch, 2003) and remark that moment matching and scenario reduction methods are not mutually exclusive. That is, a (dense) scenario tree can be generated by matching statistical properties of the historical data, and then it can be systematically reduced so that the SP becomes tractable.

This paper is organized as follows. [Section 2](#) introduces the moment matching method as a systematic method to generate scenario trees. Enhancements to each formulation of the Moment Matching Problem are also proposed. The method is illustrated via a motivating numerical example for the optimal production planning of a network of chemical plants. Moreover, approaches for reducing the scenario tree are briefly discussed. [Section 3](#) extends the methodology to the multi-stage case; the role of modeling stochastic processes is emphasized and two approaches are described based on NLP and LP statistical property matching formulations for generating multi-stage scenario trees. The approaches are illustrated with a numerical example, and conclusions are drawn in [Section 4](#).

2 Two-Stage Scenario Tree Generation

It is important to recall the role of scenario trees in Stochastic Programming (SP). Scenario trees are an *approximate discretized* representation of the uncertainty in the data (Kaut, 2003). They are based on discretized probability distributions to model the stochastic processes. The scenario trees are approximate because they contain a restricted number of outcomes in order to avoid the integration of continuous distribution functions. However, the size of the scenario trees directly impacts the computational complexity of SP models.

The concerns raised in the above paragraph have motivated the search for methodologies that can be used to systematically generate scenario trees. Two main classes of methods can be identified: scenario generation and scenario reduction. In this section, we focus on scenario generation methods, in particular, the **moment matching method**, which was

originally proposed by Høyland & Wallace (2001) and is described as follows. Given an initial structure of the tree, *i.e.* number of nodes per stage, it determines at each node the values for the random variables and their probabilities by solving a nonlinear programming (NLP) problem. The NLP problem minimizes the weighted squared error between statistical properties calculated from the outcomes or nodes, and the same properties calculated directly from the data. Thus, it is based on an L^2 -norm formulation. If the absolute deviations from the target properties are minimized as proposed by Ji *et al.* (2005), then an L^1 -norm formulation can be employed, which has the advantage that it can be cast as an LP problem. In this paper, we present a new formulation of the MMP based on the L^∞ -norm. Examples of statistical properties are the first four moments (expected value, variance, skewness, and kurtosis), covariance or correlation matrix, quantiles, etc.

In this section, we focus on two-stage problems in which the sources of uncertainty do not have a time-series effect. In Section 3, we present approaches to generating scenario trees with multiple stages where stochastic processes are the source of uncertainty.

2.1 L^2 Moment Matching Problem

In the Moment Matching Problem (MMP), the uncertain parameters of the SP model become variables in a nonlinear optimization formulation as well as the probabilities of the outcomes. The purpose of the MMP is to find the optimal values for the random variables and probabilities (see Figure 1) of a pre-specified structure for the scenario tree that minimize the error between the statistical properties calculated from the tree and the ones calculated directly from the data.

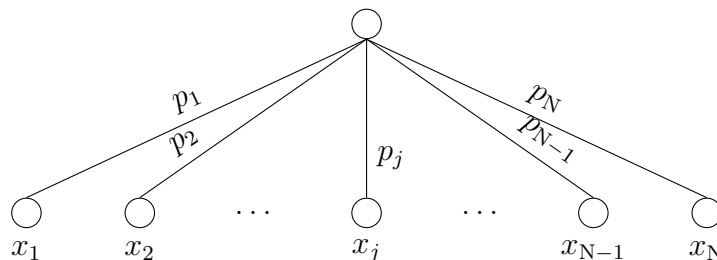


Figure 1: Two-stage scenario tree for one uncertain parameter.

In the L^2 formulation, the squared error is employed in the objective function. Hence, the NLP formulation can be generically written as follows:

$$\begin{aligned} \min_{x, p} \quad & \sum_{s \in S} w_s \cdot (f_s(x, p) - \text{Sval}_s)^2 \\ \text{s.t.} \quad & \sum_{j=1}^N p_j = 1 \end{aligned} \tag{1}$$

where x is a vector of random variables (uncertain parameters of the SP model), p is a vector of probabilities of outcomes, $s \in S$ is a statistical property to be matched (target), w_s is the weight for statistical property s , $f_s(\cdot, \cdot)$ is the mathematical expression of statistical property s calculated from the tree, Sval_s is the value of statistical property s (target value)

that characterizes the distribution of the data. Therefore, generating scenario trees via the solution of the MMP is a data-driven approach in the sense that it does not require assuming specific parametric probability distributions to model the uncertainty.

Any statistical property that somehow describes the data can be used to measure how well the scenario tree represents them. Descriptive statistics provides measures that can be used to summarize and inform us about the probability distribution of the data. Four of these measures, called moments, (Papoulis, 1991), are the following: mean or expectation, and the central moments variance, skewness, and kurtosis. The mean or expectation tells us about the *average* value in a data set, the variance is a measure of the *spread* of the data about the mean, the skewness is a measure of the *asymmetry* of the data, and the kurtosis is a measure of the *thickness* of the tails of the shape of the distribution of the data.

A more detailed definition of the L^2 MMP is as follows. The uncertain data are indexed by $i \in I$, which denotes the entity of an uncertain parameter (for example, a product). N denotes the number of outcomes per node at the second stage, $j \in J = \{1, 2, \dots, N\}$ denotes the branches (outcomes) from the root node, and $k \in K = \{1, 2, 3, 4\}$ is the index of the first four moments. The decision variables are the uncertain parameters of the stochastic programming problem, $x_{i,j}$, with corresponding probabilities of outcomes, p_j . The moments calculated from the tree are denoted by variables $m_{i,k}$ and the ones calculated from the data are denoted by parameters $M_{i,k}$. Finally, the second co-moment, *i.e.* covariance, calculated between entity i and i' from the tree and the data are denoted by $c_{i,i'}$ and $C_{i,i'}$, respectively. The L^2 MMP formulation is given as follows (see Gülpnar, Rustem, & Settergren (2004)). The goal is to generate a tree (determine the values of $x_{i,j}$ and p_j) whose properties match those calculated from the data ($M_{i,k}$ and, if applicable, $C_{i,i'}$).

(L^2 MMP)

$$\min_{x,p} z_{\text{MMP}}^{L^2} = \sum_{i \in I} \sum_{k \in K} w_{i,k} (m_{i,k} - M_{i,k})^2 + \sum_{\substack{(i,i') \in I \\ i < i'}} w_{i,i'} (c_{i,i'} - C_{i,i'})^2 \quad (2a)$$

s.t.

$$\sum_{j=1}^N p_j = 1 \quad (2b)$$

$$m_{i,1} = \sum_{j=1}^N x_{i,j} p_j \quad \forall i \in I \quad (2c)$$

$$m_{i,k} = \sum_{j=1}^N (x_{i,j} - m_{i,1})^k p_j \quad \forall i \in I, k > 1 \quad (2d)$$

$$c_{i,i'} = \sum_{j=1}^N (x_{i,j} - m_{i,1})(x_{i',j} - m_{i',1}) p_j \quad \forall (i, i') \in I, i < i' \quad (2e)$$

$$x_{i,j} \in [x_{i,j}^{\text{LB}}, x_{i,j}^{\text{UB}}] \quad \forall i \in I, j = 1, \dots, N \quad (2f)$$

$$p_j \in [0, 1] \quad \forall j = 1, \dots, N \quad (2g)$$

where the weighted squared error between the statistical properties calculated from the tree and inferred from the data is minimized in (2a), constraints (2b) ensure that the probabil-

ities of outcomes add up to 1, (2c) represent the calculation of the first moment (mean), constraints (2d) represent the calculation of higher-order central moments, constraints (2e) are the expressions for the covariance, and $w_{i,k} = w'_{i,k}/M_{i,k}^2$ and $w_{i,i'} = w'_{i,i'}/C_{i,i'}^2$, where $w'_{i,k}$ and $w'_{i,i'}$ are weights, which can be chosen arbitrarily. The bounds on the decision variables x and p are represented in constraints (2f) and (2g), respectively.

Remark 1. Skewness, $Skew$, and kurtosis, $Kurt$, are by definition normalized properties:

$$Skew_i = \frac{\sum_{j=1}^N (x_{i,j} - m_{i,1})^3 p_j}{\sigma_i^3} \quad \forall i \in I$$

$$Kurt_i = \frac{\sum_{j=1}^N (x_{i,j} - m_{i,1})^4 p_j}{\sigma_i^4} \quad \forall i \in I$$

where $\sigma_i^2 = m_{i,2}$ is the variance as defined in equation (2d) for $k = 2$. Therefore, in order to use constraints (2d) for $k > 2$ in the L^2 MMP, the statistical properties calculated from the data have to be denormalized.

Remark 2. Before solving the L^2 MMP, the number of branches or outcomes from the root node, N , is pre-specified. Høyland & Wallace (2001) suggest the rule $(|I| + 1)N - 1 \sim$ number of statistical specifications, where $|I|$ is the number of random variables. The authors also discuss potential over- and under-specification that may arise from choosing a value for N . The other inputs or parameters to the L^2 MMP are the values of the statistical properties to be matched. They directly affect the quality of the tree obtained. Hence, care should be exercised to obtain those properties in a meaningful way, so that the scenario tree effectively captures the uncertainty in the data.

Remark 3. The use of covariance or correlation information enables one to capture the *linear* dependence between multiple sources of uncertainty. More sophisticated and rigorous ways, such as copulas, to model dependency of distributions in a multivariate structure have been employed in a few papers, for instance, Sutiene & Pranevicius (2007); Kaut (2013).

Remark 4. Two theoretical concepts in scenario tree generation are stability and bias. Kaut (2003) defines two types of stability criteria: *in-sample* and *out-of-sample* stability. In-sample stability can be checked by comparing the solutions to the SP model from using different trees among each other, whereas out-of-sample stability is obtained by comparing the solutions obtained from different trees with the solution obtained from using “true” distributions. In practice, only in-sample stability can be tested as we may not know the true probability distribution of the uncertain parameters. Bias in the tree structures can be detected if the vector of solution variables are not “too similar” to the one obtained when solving the SP model with known true probability distributions. Again, this may not be possible to check in practice. More formal definitions of stability and bias can be found in Kaut (2003).

The NLP problem in equation (2) is nonconvex and its degree of nonlinearity and non-convexity increases when attempting to match higher moments. As expected, initialization plays an important role in such optimization problems. Therefore, local NLP solvers may encounter numerical difficulties and get stuck in *poor* local solutions. Systematic *multi-start*

methods can be used with local NLP solvers to help overcome the problems aforementioned by sampling multiple starting points in the feasible region and solving the NLP problem using each different starting point; however, it must be recognized that multi-start methods are not a panacea and there is no guarantee of systematically obtaining a global (or near global) solution to the MMP. Finally, deterministic global optimization solvers can also be used although at considerable computational expense.

2.2 L^1 and L^∞ Moment Matching Problems

If the absolute value of the deviations from the target moments and co-moments are minimized, then the MMP becomes an L^1 -norm model as proposed by Ji *et al.* (2005). A well-known reformulation of the nondifferentiable absolute value function in the definition of the objective function consists in splitting the variable in its argument into two non-negative variables, which correspond to the positive and negative values of the original variable.

The L^1 formulation of the MMP is then as follows. Partition the moment and covariance variables, $m_{i,k}$ and $c_{i,i'}$, respectively, into their positive and negative parts $m_{i,k}^+$, $m_{i,k}^-$, $c_{i,i'}^+$, and $c_{i,i'}^-$. Thus, the L^1 MMP is given by:

(L^1 MMP)

$$\min_{x,p} z_{\text{MMP}}^{L^1} = \sum_{i \in I} \sum_{k \in K} w_{i,k} (m_{i,k}^+ + m_{i,k}^-) + \sum_{\substack{(i,i') \in I \\ i < i'}} w_{i,i'} (c_{i,i'}^+ + c_{i,i'}^-) \quad (3a)$$

s.t.

$$\sum_{j=1}^N p_j = 1 \quad (3b)$$

$$\sum_{j=1}^N x_{i,j} p_j + m_{i,1}^+ - m_{i,1}^- = M_{i,1} \quad \forall i \in I \quad (3c)$$

$$\sum_{j=1}^N (x_{i,j} - \sum_{j'=1}^N x_{i,j'} p_{j'})^k p_j + m_{i,k}^+ - m_{i,k}^- = M_{i,k} \quad \forall i \in I, k > 1 \quad (3d)$$

$$\sum_{j=1}^N (x_{i,j} - \sum_{j'=1}^N x_{i,j'} p_{j'}) (x_{i',j} - \sum_{j'=1}^N x_{i',j'} p_{j'}) p_j + c_{i,i'}^+ - c_{i,i'}^- = C_{i,i'} \quad \forall (i, i') \in I, i < i' \quad (3e)$$

$$m_{i,k}^+, m_{i,k}^- \geq 0 \quad \forall i \in I, k \in K \quad (3f)$$

$$c_{i,i'}^+, c_{i,i'}^- \geq 0 \quad \forall (i, i') \in I, i < i' \quad (3g)$$

$$x_{i,j} \in [x_{i,j}^{\text{LB}}, x_{i,j}^{\text{UB}}] \quad \forall i \in I, j = 1, \dots, N \quad (3h)$$

$$p_j \in [0, 1] \quad \forall j = 1, \dots, N \quad (3i)$$

where the weighted absolute deviations between the statistical properties calculated from the tree and inferred from the data are minimized in (3a), constraints (3b) ensure that the

probabilities of outcomes add up to 1, (3c) attempts to match the first moment (mean), constraints (3d) represent the matching of higher-order central moments, constraints (3e) attempt to match the covariance, and $w_{i,k} = |w'_{i,k}/M_{i,k}|$ and $w_{i,i'} = |w'_{i,i'}/C_{i,i'}|$, where $w'_{i,k}$ and $w'_{i,i'}$ are weights that can be arbitrarily chosen. The bounds on the variables x , p , m^+ , m^- , c^+ , and c^- are represented by constraints (3f) – (3i).

Another way of formulating the MMP is through the minimization of the L^∞ -norm of the deviations with respect to the targets.

(L^∞ MMP)

$$\min_{x, p} z_{\text{MMP}}^{L^\infty} = \mu + \gamma \quad (4a)$$

s.t.

Constraints (3b) – (3i)

$$\mu \geq w_{i,k} m_{i,k}^+ \quad \forall i \in I, k \in K \quad (4b)$$

$$\mu \geq w_{i,k} m_{i,k}^- \quad \forall i \in I, k \in K \quad (4c)$$

$$\gamma \geq w_{i,i'} c_{i,i'}^+ \quad \forall (i, i') \in I, i < i' \quad (4d)$$

$$\gamma \geq w_{i,i'} c_{i,i'}^- \quad \forall (i, i') \in I, i < i' \quad (4e)$$

where μ and γ are scalar variables that account for the maximum deviations in the moments and covariances, respectively.

2.2.1 Linear Programming L^1 and L^∞ MMPs

The L^2 , L^1 , and L^∞ MMPs shown in equations (2), (3), and (4), respectively, are nonlinear and nonconvex due to the mathematical expressions for the moments since both probabilities and node values are decision variables. Ji *et al.* (2005) used ideas from Linear Goal Programming and proposed an LP formulation for the L^1 MMP in which only probabilities are decision variables. In this LP formulation, the node values are generally obtained via some simulation approach. For time-dependent data, such as asset returns in financial portfolio management applications, a time-series model is used to forecast future expected values and possibly higher moments. Multiple values above and below the forecast expected value can be used as the node values or outcomes in the L^1 and L^∞ LP MMP formulations and the probabilities of each outcome are left as the decision variables. In PSE applications, uncertain parameters that typically have a time component are product demand and market price.

Let $x_{i,j}$ be a parameter with the value of the uncertain parameter that can be arbitrarily chosen or calculated from some simulation procedure, for example simulation of time-series forecasting models. As long as there are at least two values, for example $x_{i,j}$ and $x_{i,j'}$, that are *symmetric* with respect to the mean, then the expected value can always be matched (see Proposition 1 in Ji *et al.* (2005)) and the L^1 LP MMP is given as follows:

(L^1 LP MMP)

$$\min_p z_{\text{LP MMP}}^{L^1} = \sum_{i \in I} \sum_{k \in K \setminus \{1\}} w_{i,k} (m_{i,k}^+ + m_{i,k}^-) + \sum_{\substack{(i, i') \in I \\ i < i'}} w_{i,i'} (c_{i,i'}^+ + c_{i,i'}^-) \quad (5a)$$

s.t.

$$\sum_{j=1}^N p_j = 1 \quad (5b)$$

$$\sum_{j=1}^N x_{i,j} p_j = M_{i,1} \quad \forall i \in I \quad (5c)$$

$$\sum_{j=1}^N (x_{i,j} - M_{i,1})^k p_j + m_{i,k}^+ - m_{i,k}^- = M_{i,k} \quad \forall i \in I, k > 1 \quad (5d)$$

$$\sum_{j=1}^N (x_{i,j} - M_{i,1})(x_{i',j} - M_{i',1}) p_j + c_{i,i'}^+ - c_{i,i'}^- = C_{i,i'} \quad \forall (i, i') \in I, i < i' \quad (5e)$$

$$m_{i,k}^+, m_{i,k}^- \geq 0 \quad \forall i \in I, k \in K \quad (5f)$$

$$c_{i,i'}^+, c_{i,i'}^- \geq 0 \quad \forall (i, i') \in I, i < i' \quad (5g)$$

$$p_j \in [0, 1] \quad \forall j = 1, \dots, N \quad (5h)$$

Likewise, the L^∞ LP MMP can be formulated as follows:

(L^∞ LP MMP)

$$\min_p z_{\text{LP MMP}}^{L^\infty} = \mu + \gamma \quad (6a)$$

s.t.

Constraints (5b) – (5h)

$$\mu \geq w_{i,k} m_{i,k}^+ \quad \forall i \in I, k \in K \quad (6b)$$

$$\mu \geq w_{i,k} m_{i,k}^- \quad \forall i \in I, k \in K \quad (6c)$$

$$\gamma \geq w_{i,i'} c_{i,i'}^+ \quad \forall (i, i') \in I, i < i' \quad (6d)$$

$$\gamma \geq w_{i,i'} c_{i,i'}^- \quad \forall (i, i') \in I, i < i' \quad (6e)$$

Obviously, it may be more advantageous to solve an LP problem instead of a nonconvex NLP problem. For multi-stage stochastic problems with time-dependent uncertain parameters, the solution strategy is much more complex when applying the NLP model instead of the LP formulation. Details are given in Sections 3.1 and 3.2.

2.3 Remarks on the MMP Formulations

Each L^p -norm formulation for the MMP produces different solutions, *i.e.* different values of probabilities, and when applicable, outcomes. This can be explained by the properties of L^p -norms of vectors. To illustrate, consider a vector $x \in \mathbb{R}^2$ where the goal is to approximate it using a point in a one-dimensional affine space A . In other words, we wish to find $\hat{x} \in A$

that it minimizes the error measured by an L^p -norm denoted by $\|x - \hat{x}\|_p$. Figure 2 shows the best approximation for the cases where $p = 1, 2$, and ∞ (Eldar & Kutyniok, 2012). The geometric shapes correspond to L^p spheres. Notice that the larger p tends to spread out the error more evenly, while smaller p leads to an error that is more unevenly distributed and tends to be sparse. This observation generalizes to higher dimensions.

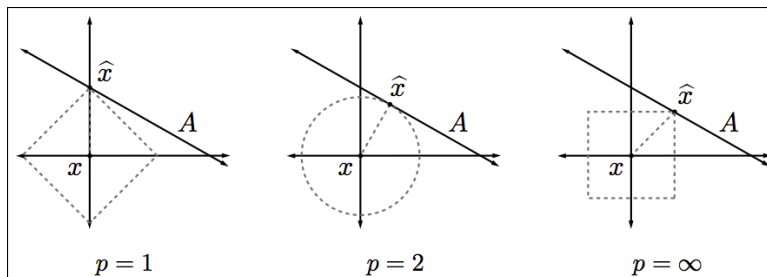


Figure 2: Best approximation of a point in \mathbb{R}^2 by a one-dimensional subspace using the L^p -norms for $p = 1, 2$, and ∞ . (Eldar & Kutyniok, 2012)

It has been our experience that it is common to have *under-specified* NLP and LP problems when only moments are matched. This is due to the fact that not enough information to be matched (statistical properties) is provided to achieve non-degenerate solutions. The consequences are that multiple choices for the node values and/or probabilities yield the same objective function value. In other words, multiple trees with the same number of nodes and having very different node values and (sometimes zero) probabilities satisfy the specifications. In addition, we observed that the Lagrange multipliers associated with all constraints in the models are zero or very small at the optimal solution obtained by local and global solvers. Moreover, the distribution obtained from solving the MMPs does not exhibit a similar shape as the distribution of the data even when up to four moments were matched.

Therefore, we propose including additional statistical properties to be matched in order to avoid solving an ill-posed problem, and to ensure that the shape of the distribution of the data is captured in the solution. This is also motivated by the fact that in certain applications it may not be practical to obtain accurate estimates of higher moments as a large amount of data is needed. Consequently, fewer moments may be matched based on their availability, while still capturing the shape of distribution of data with the scenario tree. Lastly, our numerical experiments demonstrate that the same solution vector is achieved by local and global solvers. That is, only one tree satisfies the specifications, although theoretically there is no guarantee that this property holds true due to nonconvexity in the NLP models.

An enhanced formulation – Distribution Matching Problem (DMP) – based on the MMP is proposed that not only attempts to match moments, but also the Empirical Cumulative Distribution Function (ECDF) of the data as explained in the next section.

2.4 Distribution Matching Problem

In this section, we propose enhancements to the L^2 , L^1 , L^∞ MMPs, and the L^1 and L^∞ LP MMPs in order to also match an approximation to the Empirical Cumulative Distribution

Function (ECDF) of the data. Before describing the steps of the algorithm to incorporate the ECDF information into the optimization models, some definitions are presented.

For a given random variable (r.v.) Z , the probability of Z to take on a value, say z , less than or equal to some value t is given by the Cumulative Distribution Function (CDF), or mathematically $CDF(t)$. A CDF is associated with a specific Probability Density Function (PDF), for continuous r.v.s, or Probability Mass Function (PMF), for discrete r.v.s. In order to avoid making assumptions about the distribution model, an estimator of the CDF can be used, the Empirical CDF (ECDF), which is defined as follows (van der Vaart, 1998):

$$ECDF(t) = \frac{1}{n} \sum_{i=1}^N \mathbf{1}\{z_i \leq t\} \quad (7)$$

where n is the sample size and $\mathbf{1}\{A\}$ is the indicator function of event A , that takes the value of one if event A is true, or zero otherwise. Therefore, given a value t , the ECDF returns the ratio between the number of elements in the sample that are less than or equal to t and the sample size.

Every CDF has the following properties:

- It is (not necessarily strictly) monotone non-decreasing;
- It is right-continuous;
- $\lim_{x \rightarrow -\infty} CDF(x) = 0$; and
- $\lim_{x \rightarrow +\infty} CDF(x) = 1$.

We note that most CDFs are sigmoidal. Therefore, the ECDF, as an estimator of the CDF, is also “S-shaped” in most cases. Hence, in order to incorporate the ECDF data in the optimization models in a *smooth* way, we propose fitting the Generalized Logistic Function (GLF) (Richards, 1959), also known as Richards’ Curve, or a simplified version (for instance, the Logistic Function is a special case of the GLF). The GLF is defined as follows:

$$GLF(x) = \beta_0 + \frac{\beta_1 - \beta_0}{(1 + \beta_2 e^{-\beta_3 x})^{1/\beta_4}} \quad (8)$$

where β_0 , β_1 , β_2 , β_3 , and β_4 are parameters to be estimated. When fitting the GLF to ECDF data, the GLF can be simplified by setting $\beta_0 = 0$ and $\beta_1 = 1$ as these parameters correspond to the lower and upper asymptotes, respectively. Analytical expressions for the partial derivatives of $GLF(x)$ with respect to its parameters can be derived and used to form the Jacobian matrix for least-squares fitting purposes.

The algorithm for generating a two-stage scenario tree, where the uncertain parameters have no time-series effect, by matching moments and ECDF is described as follows:

Step 1: Collect data for the (independent) uncertain parameters and obtain individual ECDF curves for each data set.

Step 2: Approximate each ECDF curve obtained by fitting the Generalized Logistic Function (GLF) or a simplified version.

Step 3: Solve a Distribution Matching Problem (DMP) defined in equations (9), (10), or (11).

Remark. We note that if a particular probability distribution family is assumed, *i.e.* a parametric approach is taken, then CDF information rather than ECDF data can be used in the DMP. This avoids the extra step of fitting a smooth curve to the ECDF data. However, very few distribution families have closed-form expressions for the CDF. Thus, approximate formulas have to be used in order to avoid evaluating integrals in the DMP.

Extended versions of the three MMP formulations for Step 3 are presented as follows. Note that since ECDF information is taken into account, we must ensure that the values of the nodes in the tree are ordered, *i.e.* order statistics. The convention adopted is the following: $x_{i,1} \leq x_{i,2} \leq \dots \leq x_{i,N}$, which is ensured via additional inequalities in each extended NLP model. Because the node values are ordered, the summation $\sum_{j'=1}^j p_{j'}$ represents the cumulative probability of the node value $x_{i,j}$.

(L^2 DMP)

$$\min_{x,p} z_{\text{DMP}}^{L^2} = z_{\text{MMP}}^{L^2} + \sum_{i \in I} \sum_{j=1}^N \omega_{i,j} \delta_{i,j}^2 \quad (9a)$$

s.t.

Constraints (2b) – (2g)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j} \quad \forall i \in I, j = 1, \dots, N \quad (9b)$$

$$x_{i,j} \leq x_{i,j+1} \quad \forall i \in I, j = 1, \dots, N-1 \quad (9c)$$

where the variables $\delta_{i,j}$ represent the deviations with respect to the ECDF data, which in turn are approximated by, for example, the GLF and is represented by the expression $\widehat{ECDF}(x_{i,j})$. In addition to minimizing the weighted square errors from matching (co-)moments, the sum of squares of the deviations $\delta_{i,j}$ is also minimized with given weights $\omega_{i,j}$ that can be chosen relative to the weights for the term involving the moments. Thus, the weights represent a trade-off between matching sample (co-)moment data and a smooth representation of the (E)CDF.

(L^1 DMP)

$$\min_{x, p} z_{\text{DMP}}^{L^1} = z_{\text{MMP}}^{L^1} + \sum_{i \in I} \sum_{j=1}^N \omega_{i,j} (\delta_{i,j}^+ + \delta_{i,j}^-) \quad (10a)$$

s.t.

Constraints (3b) – (3i)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (10b)$$

$$x_{i,j} \leq x_{i,j+1} \quad \forall i \in I, j = 1, \dots, N-1 \quad (10c)$$

where the variables $\delta_{i,j}^+$ and $\delta_{i,j}^-$ represent the positive and negative deviations with respect to the ECDF data, respectively. The expression $\widehat{ECDF}(x_{i,j})$ represents the approximation to the ECDF data obtained by, for example, fitting the GLF. The weights to the deviations are given by $\omega_{i,j}$.

(L^1 LP DMP)

$$\min_p z_{\text{LP DMP}}^{L^1} = z_{\text{LP MMP}}^{L^1} + \sum_{i \in I} \sum_{j=1}^N \omega_{i,j} (\delta_{i,j}^+ + \delta_{i,j}^-) \quad (11a)$$

s.t.

Constraints (5b) – (5h)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (11b)$$

The *constant* expression $\widehat{ECDF}(x_{i,j})$ represents the approximation to the ECDF data obtained by, for example, fitting the GLF. Note that it is required that the vector of node values is ordered, that is, $x_{i,j} \leq x_{i,j+1}$ for $j = 1, \dots, N-1$.

(L^∞ DMP)

$$\min_{x, p} z_{\text{DMP}}^{L^\infty} = z_{\text{MMP}}^{L^\infty} + \xi \quad (12a)$$

s.t.

Constraints (3b) – (3i) and (4b) – (4e)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (12b)$$

$$\xi \geq \omega_{i,j} \delta_{i,j}^+ \quad \forall i \in I, j = 1, \dots, N \quad (12c)$$

$$\xi \geq \omega_{i,j} \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (12d)$$

$$x_{i,j} \leq x_{i,j+1} \quad \forall i \in I, j = 1, \dots, N-1 \quad (12e)$$

(L^∞ LP DMP)

$$\min_p z_{\text{LP DMP}}^{L^\infty} = z_{\text{LP MMP}}^{L^\infty} + \xi \quad (13a)$$

s.t.

Constraints (5b) – (5h) and (6b) – (6e)

$$\widehat{ECDF}(x_{i,j}) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (13b)$$

$$\xi \geq \omega_{i,j} \delta_{i,j}^+ \quad \forall i \in I, j = 1, \dots, N \quad (13c)$$

$$\xi \geq \omega_{i,j} \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N \quad (13d)$$

where ξ is a scalar variable that accounts for the maximum deviations in the ECDF information.

If a parametric approach to the distribution family is taken, then the term $\widehat{ECDF}(\cdot)$ can be substituted by an exact closed-form expression, represented by $CDF(\cdot)$, or an approximate formula, denoted by $\widetilde{CDF}(\cdot)$, and no curve fitting is needed.

The distribution matching method is illustrated in the following motivating example in which the objective is to determine the optimal production plan of a network of chemical facilities or plants. For simplicity, the only uncertain parameter considered is the production yield of one facility in the network. The example demonstrates the impact that selecting a scenario tree has on the quality of the solution of the stochastic model.

2.5 Example 1: Uncertain Plant Yield

Figure 3 shows the network of the motivating example used throughout the paper. It consists of a raw material A , an intermediate product B , finished products C and D (only product D can be stored), and facilities (plants) $P1$, $P2$, and $P3$. Product C can also be purchased

from a supplier, or in the case of multiple sites, it could be transferred from another site that also produces it.

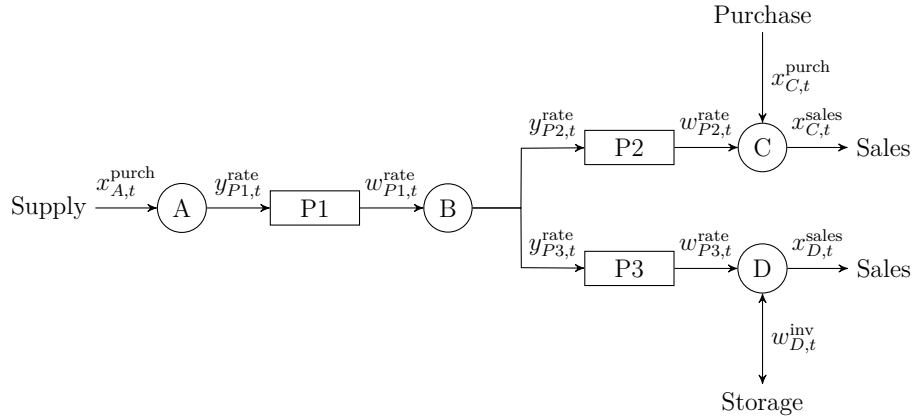


Figure 3: Network structure for the motivating Example 1.

The Linear Programming (LP) formulation has the following main elements: variables corresponding to the inlet/outlet flow rates to/from facility f in time period t , $y_{f,t}^{\text{rate}}$ and $w_{f,t}^{\text{rate}}$ respectively; production yields for each facility, θ_f ; and demands for each finished product $m \in FP$ in time period t , $\xi_{m,t}$. The deterministic multiperiod optimization model is given as follows:

$$\max w^{\text{profit}} \quad (14a)$$

$$\text{s.t. } w_{f,t}^{\text{rate}} = \theta_f y_{f,t}^{\text{rate}} \quad \forall f \in F, t \in T \quad (14b)$$

$$x_{C,t}^{\text{sales}} = w_{P2,t}^{\text{rate}} + x_{C,t}^{\text{purch}} \quad \forall t \in T \quad (14c)$$

$$w_{D,t}^{\text{inv}} = w_{D,t-1}^{\text{inv}} + w_{P3,t}^{\text{rate}} - x_{D,t}^{\text{sales}} \quad \forall t \in T \quad (14d)$$

$$w_{P1,t}^{\text{rate}} = y_{P2,t}^{\text{rate}} + y_{P3,t}^{\text{rate}} \quad \forall t \in T \quad (14e)$$

$$x_{A,t}^{\text{purch}} = y_{P1,t}^{\text{rate}} \quad \forall t \in T \quad (14f)$$

$$x_{m,t}^{\text{sales}} + \text{slack}_{m,t}^{\text{sales}} = \xi_{m,t} \quad \forall m \in FP, t \in T \quad (14g)$$

$$w_{f,t}^{\text{rate}} \leq w_{f,t}^{\text{rate,max}} + \text{slack}_{f,t}^{\text{max,cap}} \quad \forall f \in F, t \in T \quad (14h)$$

$$w_{f,t}^{\text{rate}} \geq w_{f,t}^{\text{rate,min}} - \text{slack}_{f,t}^{\text{min,cap}} \quad \forall f \in F, t \in T \quad (14i)$$

$$w_{D,t}^{\text{inv}} \leq w_{D,t}^{\text{inv,max}} \quad \forall t \in T \quad (14j)$$

$$x_{A,t}^{\text{purch}} \leq x_{A,t}^{\text{purch,max}} \quad \forall t \in T \quad (14k)$$

where constraints (14b) relate the output flows with the input flows through the yield of each facility f , constraints (14c) – (14f) represent material and inventory balances, equations (14g) represent the demand satisfaction and slack variables are employed to account for possible unmet demand, constraints (14h) – (14k) are limitations in the flows, storage, raw material

availability, and capacity violations, respectively, and the profit is calculated as follows:

$$w^{\text{profit}} = \sum_{t \in T} \left[\sum_{m \in FP} SP_{m,t} x_{m,t}^{\text{sales}} - \sum_{f \in F} OPC_{f,t} w_{f,t}^{\text{rate}} - \sum_{m \in M:MPUR=1} PC_{m,t} x_{m,t}^{\text{purch}} - \sum_{m \in M:MINV=1} IC_{m,t} w_{m,t}^{\text{inv}} - \sum_{m \in FP} PEN_{m,t} slack_{m,t}^{\text{sales}} - \sum_{f \in F} PEN_{f,t} (slack_{f,t}^{\text{max,cap}} + slack_{f,t}^{\text{min,cap}}) \right]$$

where $SP_{m,t}$ is the selling price of material m in period t , $OPC_{f,t}$ is the operating cost of facility f in period t , $PC_{m,t}$ is the purchase cost of material m in period t , $IC_{m,t}$ is the inventory cost of material m in period t , and $PEN_{m,t}$ denotes the penalty associated with unmet demand.

Consider historical data showing the variability of the production yield of facility $P1$, θ_{P1} , with 120 data points, which represent monthly records of θ_{P1} for a period of ten years. The distribution of θ_{P1} is depicted in a histogram as shown in Figure 4. Only the first two moments and ECDF data were estimated from the randomly generated production yield values. The simplified GLF ($\beta_0 = 0$ and $\beta_1 = 1$) fit to ECDF data and the estimated parameters are shown in Figure 5. Details of the procedure for generating the historical data for θ_{P1} , fitting the simplified GLF, and the remaining parameters for the production planning model are given in Appendix A.

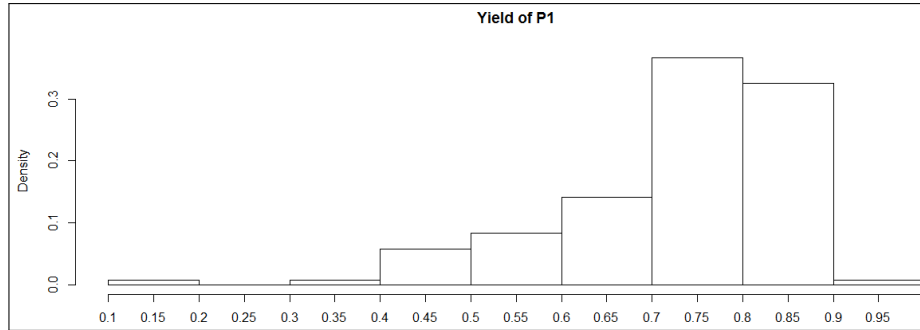


Figure 4: Distribution of the historical data for the production yield of facility $P1$.

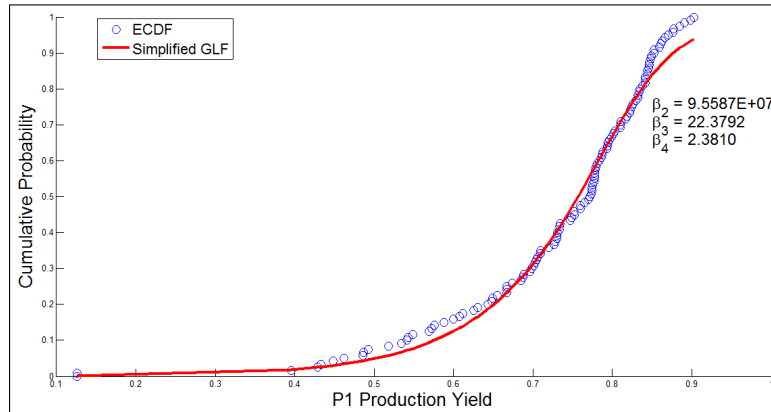


Figure 5: ECDF data of the production yield of facility $P1$ fitted by a simplified GLF.

To simplify the analysis, we model this multiperiod production planning problem as a Two-Stage Stochastic Programming (TSSP) problem. There are four time periods that correspond to a quarterly production plan problem over the course of one year time horizon. The first stage or *here-and-now* variables are all the model variables in the model at the first time period, $t = 1$, whereas the second stage or *wait-and-see* variables are all the model variables at the remaining time periods, $t > 1$. All the DMPs and the deterministic equivalent of the TSSP models are implemented in AIMMS 3.13 (Roelofs & Bisschop, 2013). The DMPs were solved with IPOPT 3.10.1 using the Multi-Start Module in AIMMS and the TSSP model was solved with Gurobi 5.1. The model sizes are small; therefore, CPU times are not reported. In the DMPs, the yield variables were bounded below by the minimum data point, and above by the maximum data point.

Five scenarios are selected for the two-stage scenario tree. Two approaches are compared: heuristic and DMP, which includes the optimization models described in Subsection 2.4. The heuristic approach represents an arbitrary way to construct a scenario tree that does *not* consider the distribution of the historical data. From minimum and maximum data values (*e.g.*, production yield that can vary between 0 and 1), their arithmetic mean is calculated (center node) and the values of the other nodes are calculated by fixed deviations of $\pm 20\%$ and $\pm 40\%$ from the mean node. Therefore, the tree in this example has five nodes in the second stage. Also, the probabilities are arbitrarily chosen. Notice that by not visualizing the distribution of the uncertain parameter, choice of outcomes and their probabilities may not satisfactorily characterize the shape of the distribution of the actual data. In other words, the heuristic scenario tree does not represent the actual problem data and the production plan obtained may not be very meaningful. The DMP approach calculates the probabilities (both LP and NLP formulations) and values of the nodes (only NLP formulations) in order to match statistical properties that describe the distribution of the yield data. The targets for the DMPs include the first two moments and ECDF data. Figure 6 shows the probabilities and yield values obtained for the five-scenario tree in each approach.

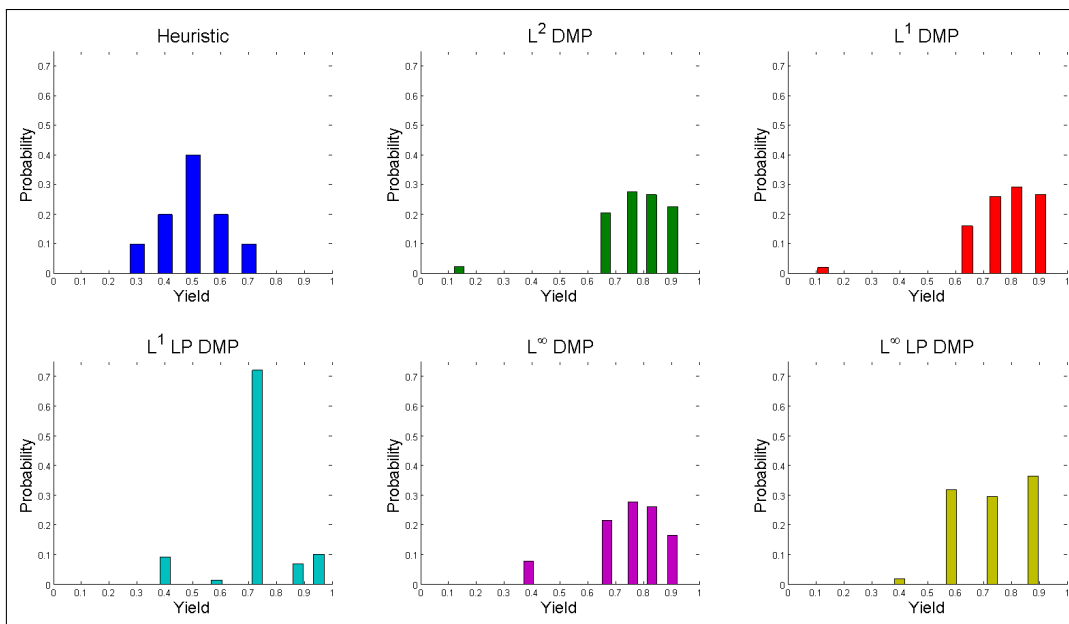


Figure 6: Probability profiles for the heuristic and optimization-based (DMP) approaches in Example 1. For reference, a histogram of the uncertain data is depicted in Figure 4.

Note that despite attempting to match only the first two moments, the additional ECDF information allows for the probability profiles obtained by solving the L^2 DMP, L^1 DMP and L^∞ DMP formulations to satisfactorily capture the shape of the distribution of the uncertain parameter. The yield distribution as shown in Figure 4 is skewed to the right, which results in higher probabilities assigned to node values that are slightly higher than the mean yield (0.7301). Such characteristic is not captured in the heuristic approach. Thus, it does not satisfactorily represent the actual data. It was observed that the probabilities obtained with the L^1 LP DMP and L^∞ LP DMP formulations were strongly dependent on the node values chosen and this fact will affect the remaining results shown below. The objective function value of the three DMP formulations are shown in Table 1. Note that the extra degrees of freedom associated with considering the node values as variables (NLP formulations) resulted in smaller deviations in the matching procedure for the choices of weights (see Appendix A).

Table 1: Objective function values of the DMP formulations in Example 1. The values represent the error of matching the statistical properties.

Model	Objective Function
L^2 DMP	0.0081
L^1 DMP	0.1453
L^1 LP DMP	0.5200
L^∞ DMP	0.0918
L^∞ LP DMP	0.2317

Table 2 shows the optimal expected profit of the stochastic production planning model in equation (14). The relatively low expected profit by using the heuristic approach can

be explained due to high probabilities placed on production yields below the mean of the actual yield data. In other words, the scenario tree in the heuristic approach is *pessimistic* for the values chosen for the production yield of facility $P1$. Ultimately, the tree in the heuristic approach is an inaccurate representation of the yield data. However, note that the magnitude of the expected profit of the TSSP problem is not an assessment of the quality of each solution with respect to the “true” solution that would be obtained if the true distributions were known and were not approximated by finite discrete outcomes. The LP deterministic equivalent two-stage stochastic program has 273 constraints, 305 variables, 832 nonzeros, and was solved in not more than 0.02 seconds for all approaches.

Table 2: Expected profit of the production planning model in Example 1 using the scenario trees from two approaches.

Approach	Expected Profit [\$]
Heuristic	65.43
L^2 DMP	73.23
L^1 DMP	73.70
L^1 LP DMP	76.48
L^∞ DMP	77.05
L^∞ LP DMP	76.39

Figures 7 and 8 show the different production plans obtained for each approach. Specifically, the solution obtained with the heuristic approach predicted higher overall inventory levels of product D for the time horizon under consideration. Moreover, using the tree obtained with the heuristic approach incurred higher purchase amounts of product C in every period in the time horizon, which can be explained by the fact that the scenario tree in that approach is constructed around a lower mean than the actual mean estimated from the data.

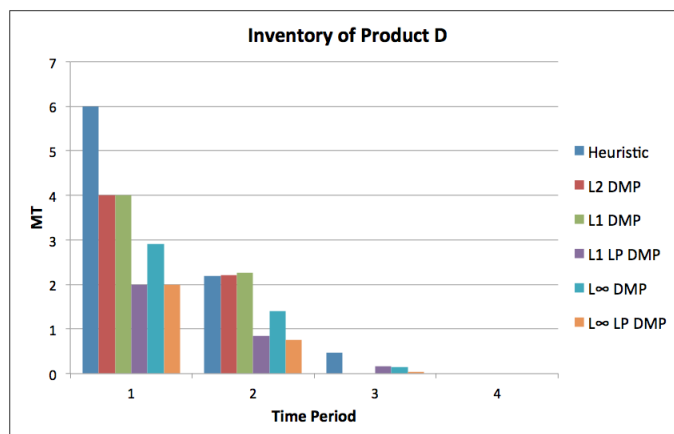


Figure 7: Optimal inventory levels of product D from using the scenarios obtained from heuristic and DMP approaches.

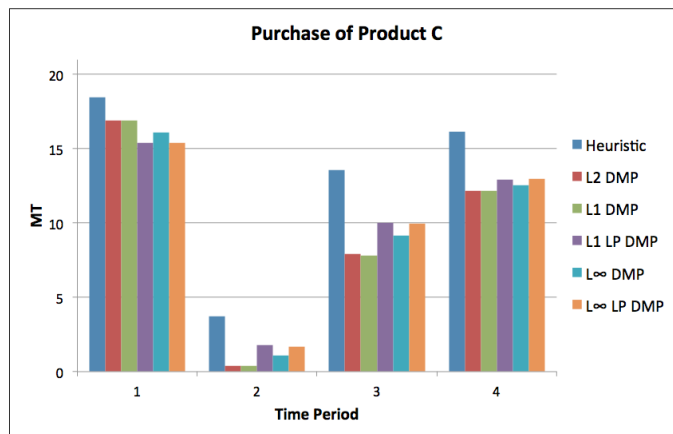


Figure 8: Optimal purchase amounts of product C from using the scenarios obtained from heuristic and DMP approaches.

Finally, the quality of the stochastic solutions was assessed using a simulation-based Monte Carlo sampling scheme that provides statistical bounds on the optimality gap (Bayrakshan & Morton, 2006). The optimality gap is defined as the difference between an approximation of the “true” stochastic solution (very large tree to approximate the continuous distribution of the yield) and the candidate stochastic solution. In this context, a candidate stochastic solution refers to the first-stage decisions of the TSSP when using the scenario trees from either heuristic or DMP approaches. The Multiple Replications Procedure (MRP) was used with twenty replications, and each replication contains one hundred independent scenarios. In each replication, the gap between the approximate true solution and the candidate solution is calculated. Table 3 shows the average gap of all replications plus one-sided confidence intervals for 95% confidence. The results suggest that the stochastic production planning solutions obtained by using the trees generated by the L^2 DMP and L^1 DMP approaches are closer to the true solution as seen from the small optimality gap. Note that since the historical data were artificially generated, *i.e.* the data-generating mechanism (distribution) is known (see Appendix A), a Monte Carlo sampling strategy can be used.

Table 3: Average value and upper bound of the optimality gap of the stochastic production planning model in Example 1.

Approach	Avg Gap [\$]	Upper Bound [\$]
Heuristic	0.21	0.29
L^2 DMP	0.04	0.10
L^1 DMP	0.04	0.11
L^1 LP DMP	0.70	0.91
L^∞ DMP	0.37	0.50
L^∞ LP DMP	0.70	0.91

The results in Table 3 clearly show that selecting the scenario tree as an input to the stochastic optimization model is crucial to obtain a meaningful solution of an SP formulation. The distribution matching method is a scenario generation method that allows creating

scenario trees that satisfactorily represent the distribution of the data, so that the decisions made with the stochastic optimization model are supported by factual probabilistic information.

2.6 Reducing the Scenario Tree

Two methods have been commonly used in the literature to remove scenarios by aggregating neighboring nodes in a scenario tree. The **scenario reduction method**, originally proposed by Dupačová, Gröwe-Kuska, & Römisch (2003), and later improved by Heitsch & Römisch (2009), is a heuristic that attempts to generate a tree with pre-specified number of scenarios that is the closest to the “original” distribution according to a probability metric. The intuitive idea is to find a subset of the original set of scenarios of prescribed cardinality that has the shortest distance to the remaining scenarios. Feng & Ryan (2013) modified that method to also account for what the authors call “key” first-stage decisions in the aggregation of scenarios in addition to probability metrics.

Another common approach to reduce the number of scenarios comes from **clustering methods** in data mining (Hastie, Tibshirani, & Friedman, 2009). In particular, some authors have used variants of the k -means clustering algorithm, where the main goal is to group or aggregate paths in the scenario tree that are “near” to each other according to some distance metric that is minimized. For instance, Xu, Chen, & Yang (2012) proposed a k -means algorithm that generates a scenario tree from a fan-like tree that not only groups paths that are near in a probabilistic sense, but also accounts for inter-stage dependency of the data.

3 Multi-Stage Scenario Tree Generation

The MMPs in equations (2) – (6) and DMPs in equations (9) – (13) can be applied to both two-stage and multi-stage cases. When generating multi-stage scenario trees, a statistical property matching problem is solved at every node of the tree except at the leaf or terminal nodes. Multi-stage scenario trees can be viewed as a group of two-stage subtrees that are formed by branching out from every node except the leaf nodes. The complication in generating multi-stage scenario trees with interstage dependency lies in the fact that the moments calculated for each path into future stages are dependent on the previous *states* or nodes present in each path. Therefore, prediction of future events (time series forecasting) must be combined with property matching optimization as will be described below.

For stochastic processes, such as time-series data of product demands, statistical properties can be estimated through forecasting models that take into account information that is *conditional* on past events. Appendix ?? contains more details of using time series forecasting models to estimate the statistical properties.

Time series forecasting models play an essential role in generating scenario trees when there is a time-series effect in the uncertain parameters. Briefly, mathematical models are fit to the historical data and their predictive capabilities provide *conditional* (co-)moments of the uncertain parameters at future stages. The moments are conditional on past events. That is, they take into account the serial dependency of the time-stamped observed values of the uncertain parameters. Hence, the statistical moments are supplied to the property

matching optimization models at each non-leaf node and the scenario tree is consistently generated. Moreover, simulation of the time series models can be used to generate data of which an ECDF can be constructed and approximated with a smooth function, such as the GLF or a simplified version (see [Subsection 2.4](#)).

In the next two sections, we present two *sequential* solution strategies for generating a multi-stage scenario tree: *NLP Approach* and *LP Approach*. We focus on the DMP formulations, where L^2 DMP, L^1 DMP and L^∞ DMP are nonlinear (both node values and probabilities are variables), whereas L^1 LP DMP and L^∞ LP DMP are linear (only probabilities are variables). Each approach comprises two main steps, **forecasting** and **optimization**, which are summarized below:

Forecasting Step: After successfully fitting a time series model to the data, forecast future values.

- Input: observed data represented by the nodes
- Output: conditional moments and ECDF information to be matched in the optimization step

Optimization Step: Solve a DMP at a given node in the tree.

- Input: conditional moments estimated by the forecasting step
- Output: probabilities of outcomes, and if using the NLP approach, values of the nodes

Remark 1. Conditional (co-)moments are readily available via forecasting. ECDF information can be obtained through simulation of time series models, and a brief overview is given in ?? in Appendix ?. If a particular family of distribution is assumed for the forecast data, then the CDF or an approximate expression can be used instead as illustrated in Example 2 ([Subsection 3.3](#)).

Remark 2. Instead of a forecasting step, some authors have used a simulation step to generate the targets (conditional moments) to the optimization step and/or the values of the nodes in the scenario tree. For example, [Høyland, Kaut, & Wallace \(2003\)](#) proposed a heuristic method to produce a discrete joint distribution of the stochastic process that is consistent with specified values of the first four marginal moments and correlations. In addition, due to the independence of the optimization problems to be solved at each node of a given stage, there is an opportunity for parallel algorithms to speed up the solution process ([Beraldi, De Simone, & Violi, 2010](#)).

3.1 NLP Approach

A general solution strategy for generating a multi-stage scenario tree using the NLP formulations of the DMP consists of alternating the two main steps described above in a shrinking-horizon fashion, *i.e.* marching forward in time, node-by-node and stage-by-stage until the end of the time horizon. [Figure 9](#) depicts the sequence of steps in the approach. The black dots (past region) in each subfigure correspond to historical data of the uncertain parameter under consideration, the blue line represents the time series model used to make predictions

of the stochastic process, the red dots (future region) are the possible future states that the stochastic process will visit, and the grey shaded area surrounding the red dots denotes the estimated prediction confidence limits for a given significance level of α , *i.e.* $\alpha = 0.05$ indicates 95% confidence. Note that by connecting parent nodes to their descendants (from left to right), a scenario tree is obtained.

The algorithm can be stated as follows.

Step 0: Start at the root (“present”) node whose value is known. Set it as the current node.

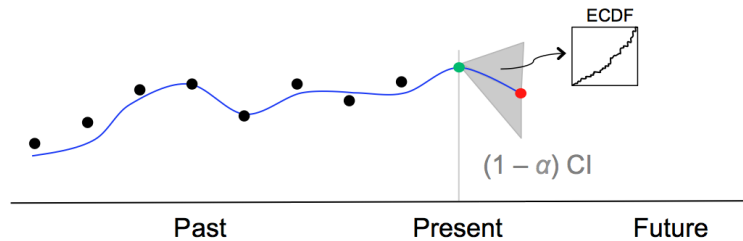
Step 1: If not the last stage in the time horizon, then perform a one-step-ahead forecast from the current node to estimate conditional moments.

Step 2: Simulate the time series model including observations up to the current node, construct the ECDF curve, and approximate it by a smooth function, such as the GLF or a simplified version.

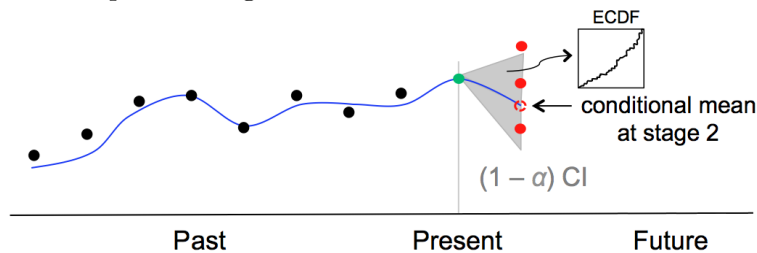
Step 3: Solve a nonlinear DMP to determine the node values and their probabilities for the next stage.

Step 4: For each node determined, set it as the current node and go to **Step 1**.

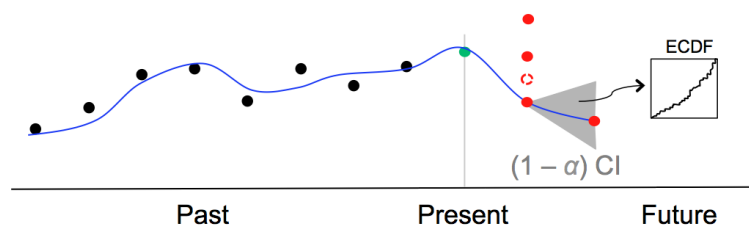
In [Figure 9](#), the blue curve represents the time series model, the black dots represent past or historical data, the green dot is the current or present state, and the red dots are the future states. For demonstration purposes, [Figure 9\(c\)](#) only shows the forecasting step for the bottom node generated in the first optimization step. Note that some nodes may lie outside the confidence interval predicted by the forecasting step; this allows more extreme events to be captured in the scenario tree, which in turn subject the stochastic programming problem to riskier scenarios and may lead to more “robust” solutions.



(a) First one-step-ahead forecasting step to predict the most likely value of the stochastic process in the next stage as well as possible higher moments and distribution information from simulation.



(b) Optimization step to calculate probabilities and nodes for the next stage. Optionally, the node corresponding to the conditional mean may be fixed in the DMP.



(c) One-step-ahead forecasting step from a given node obtained in the optimization step before. Repeat these steps for every node generated in every stage until the end of the time horizon considered.

Figure 9: Alternating forecasting and optimization steps in generating multi-stage scenario trees using the *NLP Approach*. CI denotes the confidence interval estimated at each forecast and ECDF means Empirical Cumulative Distribution Function.

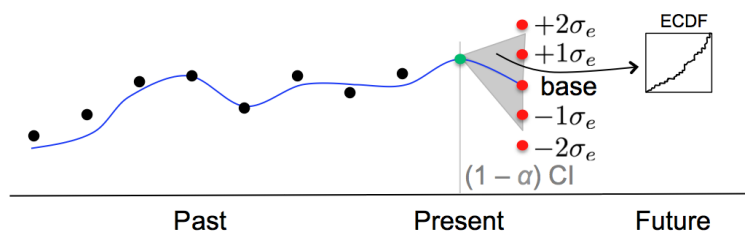
The complexity in implementing this approach in practice is the communication between the forecasting and the optimization steps at every non-leaf node in the tree. On the other hand, the approach using the LP formulations of the property matching problems only alternates between the forecasting and optimization steps once. The next section contains our proposed approach, and we note that there are variants in the literature that for instance use clustering algorithms as discussed in [Subsection 2.6](#).

3.2 LP Approach

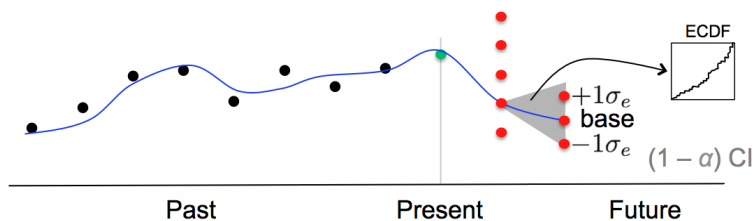
The only decision variables in the LP formulation in equation (11) are the probabilities of the outcomes. Therefore, if the node values are known in advance, then a single optimization problem can be solved for the entire tree to compute their probabilities. Thus, the approach has only two steps: (1) the forecasting step generates the nodes plus the statistical properties to be matched, and (2) an LP DMP is solved for all non-leaf nodes simultaneously. The

optimization step is a straightforward solution of an LP problem, whereas the forecasting step contains elements that are particular to a specific strategy.

The strategy for the forecasting step proposed in this paper is shown in Figure 10. As shown in Figure 10(a), after performing a one-step-ahead forecast from the present node to the base or most likely node in the second stage, additional nodes are created by adding and subtracting multiples of the standard error of the forecast to the base node. The number of additional nodes above and below the base node is chosen *a priori*. In practice, near-future stages may be more finely discretized than far-future stages, since the prediction is less accurate the further into the future it is made. For ease of exposition, Figure 10(b) shows the forecast from the second to the third stage of one of the nodes created in the second stage. The process is repeated for every node in every stage, except the last one of the time horizon considered.



(a) First one-step-ahead forecasting step to predict the most likely value of the stochastic process in the next stage. Create new nodes by adding and subtracting multiples of the standard error, σ_e , of the forecast to the base node.



(b) For each node created, perform a one-step-ahead forecast and create new nodes. Repeat the process until the end of the time horizon considered.

Figure 10: Proposed forecasting step in generating multi-stage scenario trees using the *LP Approach*. σ_e , CI, and ECDF denote the standard error, confidence interval estimated at each forecast, and the Empirical Cumulative Distribution Function, respectively.

In summary, the main difference between the *NLP Approach* and the *LP Approach* is that, in the latter, the forecasting and optimization steps alternate only once as all the nodes or outcomes of the tree are created in the forecasting step, and then the optimization step is executed to compute the probabilities.

The next example demonstrates how the two approaches can be used to generate a multi-stage scenario tree when product demand is uncertain.

3.3 Example 2: Uncertain Product Demands

Consider the same network depicted in [Figure 3](#) and the deterministic multiperiod production planning model defined in (14). In this case, the product demands of C and D are the uncertain parameters. The planning horizon is one year, which is divided into time periods of quarters. Quarterly historical demand data are given from the years of 2008 to 2012. Thus, the frequency (number of observations per year) of the time series is four. The objective is to obtain the optimal quarterly production plan for the year of 2013. As in Example 1, all optimization models were implemented in AIMMS 3.13. The NLP problems were solved with IPOPT 3.10.1 using the multi-start module in AIMMS with 30 sample points and 10 selected points in each iteration. All LP models were solved with CPLEX 12.5. In the DMPs, the demand variables were bounded below by half the minimum and above by double the maximum historical demand data points.

A common approach for deciding the structure of multi-stage trees is to select more outcomes per node in earlier stages than in later stages, since the uncertainty in the forecasts is much higher in the latter. Thus, it is more reasonable to select a finer discretization in earlier stages. It is then decided that the multi-stage scenario tree has the following structure: 1-5-3-1, which means that the second quarter has five outcomes, the third quarter has three outcomes for each outcome in the second quarter, and the fourth quarter has only one outcome for each outcome of the third quarter, thus, the scenario tree has 15 scenarios as seen in [Figure 11](#). As in Example 1, a heuristic approach is compared with the optimization-based DMPs to obtain scenario trees. We consider uncertainty in the demand of both products C and D . The tree for each individual product demand is obtained as follows. The center or base node at a given quarter is the arithmetic average of the corresponding quarter of previous years, and the remaining nodes above and below the base node are obtained by fixed deviations. Therefore, the node values *ignore* the serial dependence and time-series effects in the data. The individual heuristic trees for products C and D were combined into a single tree with the same structure (1-5-3-1) by overlapping the outcomes for each stage as shown in [Figure 11](#). Probabilities of outcomes were arbitrarily chosen and are symmetric with respect to each base node.

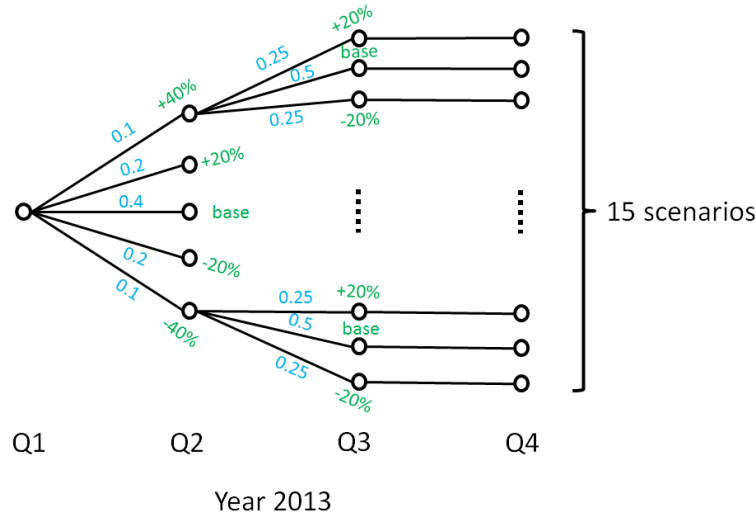


Figure 11: Heuristic scenario tree for the demand of products *C* and *D*. The percentage deviations are computed based on each base node. The values above and below the arcs are arbitrarily chosen probabilities.

Figure 12 shows the time series demand data for products *C* and *D*. The time series model that best fits the data (see Appendix B for details) is fixed in subsequent forecasts. That is, when executing an approach no refitting is performed prior to forecasting. The root node of the tree, which is the node value at the first quarter in 2013 or “Q12013”, is forecast and assumed to have probability of one. The constant variance for the demand of products *C* and *D* are estimated to be $1.65t^2$ and $1.14t^2$, respectively, and the properties matched are first two moments, covariance, and CDF information.

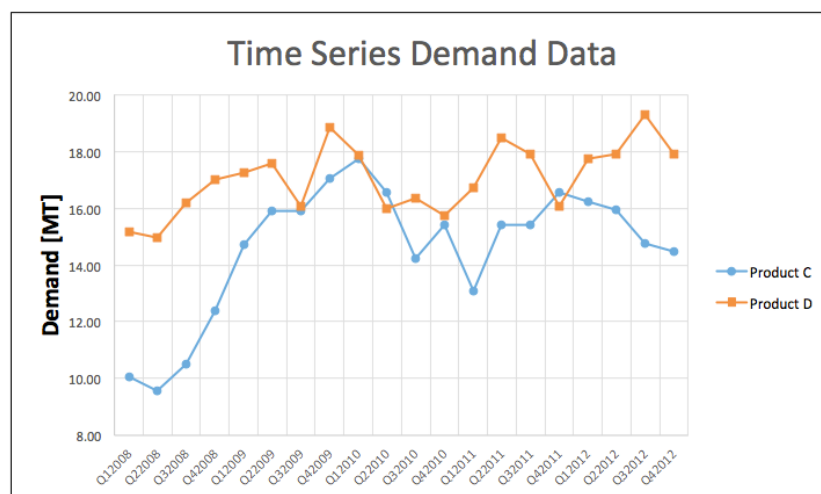


Figure 12: Time series data of the demand of products *C* and *D*.

Since the demand data are fitted to a linear Gaussian model (ARIMA), the forecasts are expected to follow normal distributions. Therefore, an expression for the Cumulative Distribution Function (CDF) of a normal distribution with mean μ and standard deviation

σ can be used in the constraints involving $\widehat{ECDF}(\cdot)$. In particular, the CDF of a normal distribution can be written in terms of the error function as follows (Abramowitz & Stegun, 1965):

$$CDF(x)^{\text{Normal}} := \Phi\left(\frac{x - \mu}{\sigma}\right) = \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{x - \mu}{\sigma\sqrt{2}}\right) \right]$$

where $\operatorname{erf}(\cdot)$ is the error function defined as the following integral:

$$\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$$

Hence, constraints (9b) can be replaced with,

$$\Phi\left(\frac{x_{i,j} - M_{i,1}}{\sqrt{M_{i,2}}}\right) - \sum_{j'=1}^j p_{j'} = \delta_{i,j} \quad \forall i \in I, j = 1, \dots, N$$

constraints (10b) and (12b) can be substituted by,

$$\Phi\left(\frac{x_{i,j} - M_{i,1}}{\sqrt{M_{i,2}}}\right) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N$$

and finally constraints (11b) and (13b) are rewritten as,

$$\Phi\left(\frac{x_{i,j} - M_{i,1}}{\sqrt{M_{i,2}}}\right) - \sum_{j'=1}^j p_{j'} = \delta_{i,j}^+ - \delta_{i,j}^- \quad \forall i \in I, j = 1, \dots, N$$

AIMMS offers a native, numerical approximate implementation of the error function, which can be directly used in the implementation of the constraints in the DMPs.

Exclusively for the LP DMPs, it was observed that additional constraints on the probabilities were necessary in order to enforce a normal-like profile, *i.e.* probabilities monotonically decrease from the center node outward. The additional constraints are given below and are equivalent to the ones proposed by Ji *et al.* (2005).

$$\begin{aligned} p_j &\geq p_{j'} & \forall j = \left\lceil \frac{N}{2} \right\rceil, \dots, N, j' > j \\ p_j &\geq p_{j'} & \forall j = 1, \dots, \left\lfloor \frac{N}{2} \right\rfloor, j' < j \end{aligned}$$

For illustration purposes, the scenario trees obtained with NLP and LP approaches are shown in Figure 13 (L^2 DMP) and Figure 14 (L^∞ LP DMP), respectively. For the *NLP Approach*, the node values in the fourth time period correspond to the conditional means obtained via forecasting, *i.e.* no optimization was needed as only one outcome was considered. It should be noted that the total time for solving the six NLPs with multi-start for the tree in Figure 13 was 12.35 seconds, while the LP in Figure 14 took 0.02 seconds.

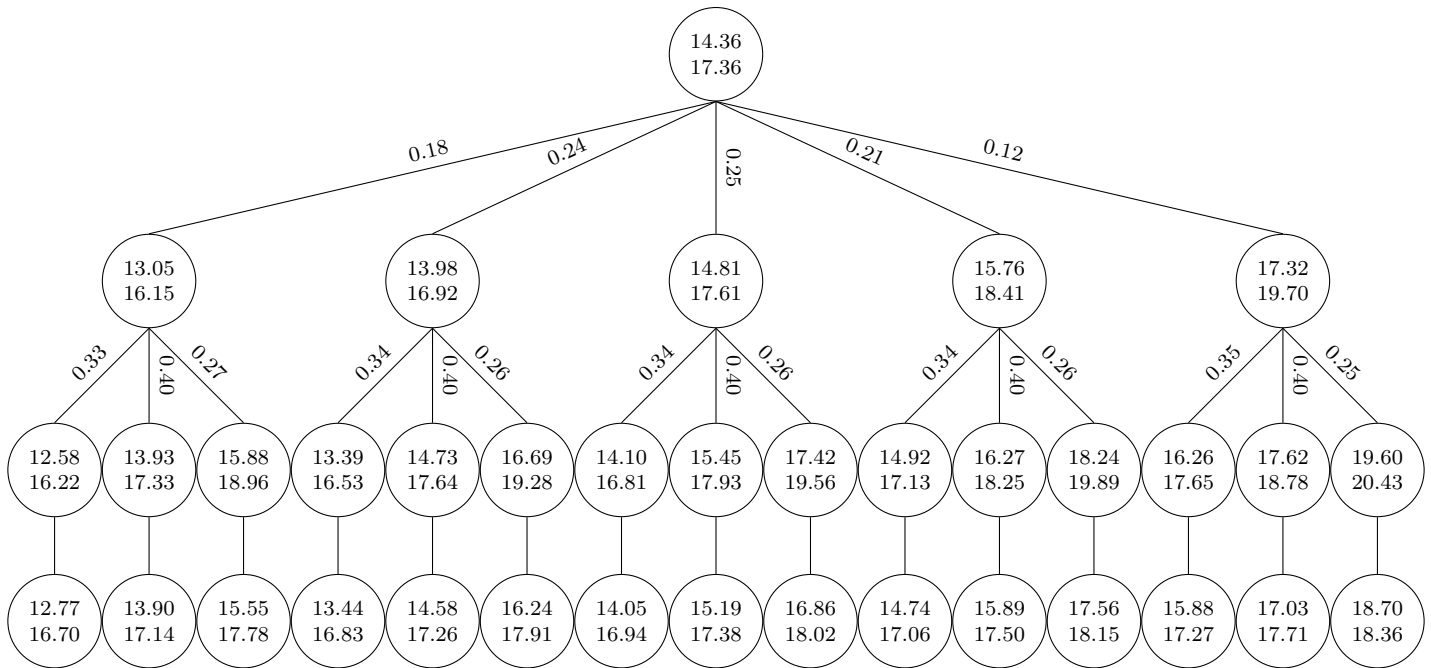


Figure 13: Scenario tree obtained with the *NLP Approach* (L^2 DMP) for Example 2. Top and bottom values inside each node are the calculated demands of products *C* and *D*, respectively.

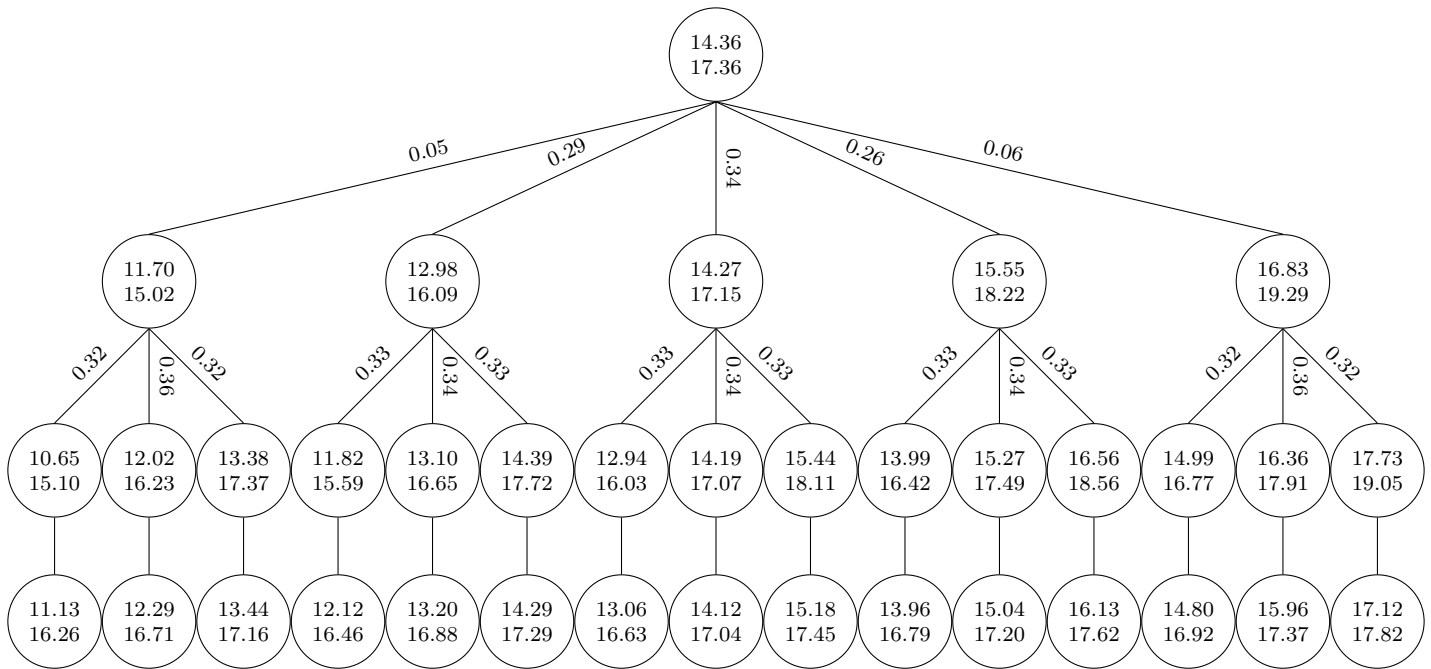


Figure 14: Scenario tree obtained with the *LP Approach* (L^∞ LP DMP) for Example 2. The node values are obtained via forecasting and the probabilities are calculated via optimization. Top and bottom values in each node are the demands of products *C* and *D*, respectively.

The optimal expected profit of the production planning model by using the scenario

tree of the proposed approach as the input, and by solving the deterministic equivalent of the multi-stage stochastic programming model is shown in Table 4. The heuristic approach underestimates the expected total profit when compared to the NLP DMPs. Again, we note that the scenario probabilities and the solution obtained with the LP DMPs is greatly affected by the node values chosen. The LP deterministic equivalent of the multi-stage stochastic program has 613 constraints, 685 variables, 1,872 nonzeros, and was solved in less than 0.02 seconds for all approaches.

Table 4: Expected profit in Example 2 using the scenario trees from heuristic and optimization-based approaches.

Approach	Expected Profit [\$]
Heuristic	79.95
L^2 DMP	82.39
L^1 DMP	82.65
L^1 LP DMP	80.04
L^∞ DMP	82.41
L^∞ LP DMP	80.03

Figures 15 and 16 show the different production plans obtained for each approach at each quarter. Specifically, the solution obtained with the heuristic approach predicted higher overall inventory levels of product D for the time horizon under consideration. Moreover, the solution using the heuristic tree shows very different average flowrates out of plant $P2$ compared to the ones obtained using the DMP formulations. In real life terms, the production quota for a plant affects lower-level operability decisions, such as scheduling and control.

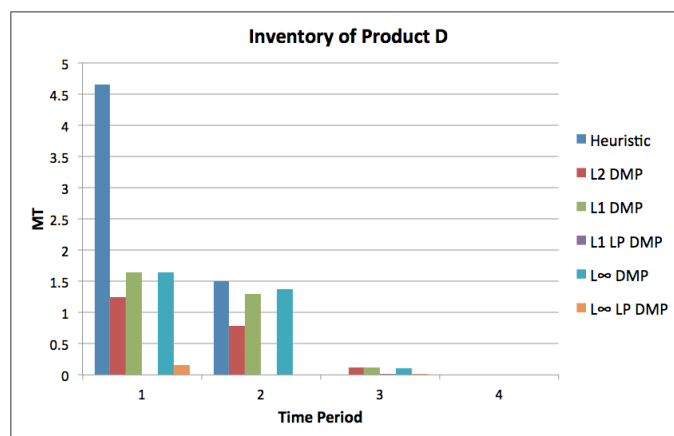


Figure 15: Optimal inventory levels of product D from using the scenarios obtained from heuristic and DMP approaches.

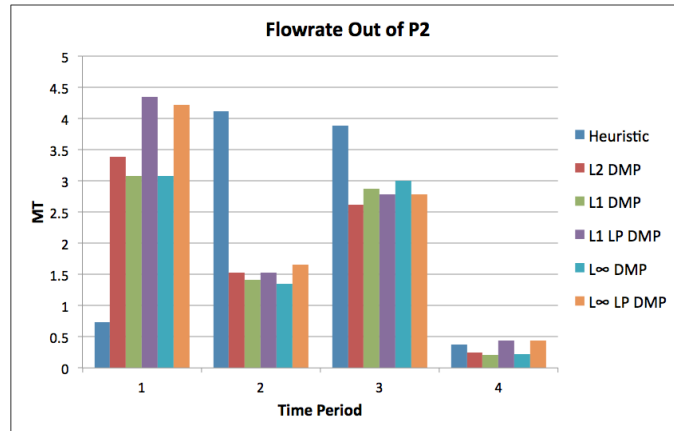


Figure 16: Optimal flow rates out of plant P_2 from using the scenarios obtained from heuristic and DMP approaches.

Finally, similarly to Example 1, the quality of the stochastic solutions was assessed using a simulation-based Monte Carlo sampling scheme that provides statistical bounds on the optimality gap (Chiralaksanakul & Morton, 2004). The optimality gap is calculated using a tree-based estimator of the lower bound (candidate solution, maximization problem) and the approximate true solution (upper bound estimator, maximization problem). The simulated trees have the structure 1-10-10, which amounts to one thousand scenarios. All subtrees are generated by simulating ARIMA processes (see ?? in Appendix ??). Ten replications of the algorithm (see Procedure \mathcal{P}_2 in the original paper) were performed to obtain the confidence interval on the gap. Table 5 shows the one-sided confidence intervals for 95% confidence. Note that by modeling the demand time series as ARIMA processes, the data-generating mechanism is known, and Monte Carlo sampling can be performed by simulating the ARIMA models (see ?? in Appendix ??).

Table 5: Average value and upper bond of the optimality gap of the stochastic production planning model in Example 2.

Approach	Avg Gap [\$]	Upper Bound [\$]
Heuristic	2.03	2.16
L^2 DMP	0.92	1.00
L^1 DMP	0.93	1.01
L^1 LP DMP	1.12	1.35
L^∞ DMP	0.93	1.01
L^∞ LP DMP	1.04	1.22

Note that the confidence interval of the gaps obtained for all the DMP formulations are lower than the one obtained for the heuristic approach, which indicates that the scenario trees generated via the optimization-based procedure are good approximations of the “true” distribution. In addition, they contain correlation information between the demands of the two products, thus improving the characterization of the uncertainty.

4 Conclusions

In this paper, we have described a systematic method for scenario tree generation that can be used in optimization and simulation solution strategies in Process Systems Engineering (PSE) problems. The distribution matching method is based on an optimization model that can be used to calculate the probabilities and values of outcomes in a scenario tree. This is accomplished by attempting to match statistical properties, such as (co-)moments and Empirical Cumulative Distribution Function (ECDF) information, calculated from the tree to the ones estimated from the data. The motivation behind this approach is to ensure that the scenarios considered in a stochastic programming framework correspond to the behavior observed in historical and forecast data, as opposed to assigning arbitrary values and probabilities to the nodes in the scenario tree. Therefore, characterizing the uncertainty adds value to the solution of the stochastic model used for decision-making.

For multi-stage scenario tree generation, we presented two approaches that can be used in conjunction with time series forecasting or simulation. The *NLP Approach* is composed of two alternating steps: the forecasting or simulation step computes conditional (co-)moments and ECDF information to be matched, and the optimization step determines the probabilities and values of the outcomes in order to match those properties. The two steps are alternated until the prescribed time horizon is completed. The *LP Approach* also contains the same two steps, but each step is performed only once. First, the forecasting or simulation step generates all the nodes in the tree and computes the conditional (co-)moments and ECDF information, and then the optimization step calculates the probabilities of outcomes.

The quality of the solution of the stochastic programming problems was assessed via simulation-based Monte Carlo sampling methods. It was shown that the proposed scenario tree generation methods provide good-quality solutions, and provide a systematic approach for handling multiple sources of uncertainty, thus generating more compact trees with covariance or correlation information.

Acknowledgment

The authors gratefully acknowledge financial support from The Dow Chemical Company. They also are grateful to Alex Kalos for the fruitful discussions regarding the incorporation of (E)CDF information to the moment matching method, and to the Advanced Analytics team, especially Joe Czyzyk, with regards to time series modeling and forecasting.

Appendix A Data for Example 1

The production yield data were randomly generated using the R programming language version 3.0.1 (R Core Team, 2013). In particular, the library `PearsonDS` (Becker & Klößner, 2013) was used to generate 120 random numbers sampled from a Pearson distribution with given mean, variance, skewness, and kurtosis. The four moments of the generated data were calculated using functions of the library `e1071` (Meyer *et al.*, 2012). The code in Listing A.1 can be used to reproduce the yield data in Example 1.

```

library(PearsonDS)
library(e1071)
moments <- c(mean=0.7, variance=0.02, skewness=-1, kurtosis=4)
set.seed(1234)
v <- rpearson(10*12, moments=moments)
data.min <- min(v)
data.max <- max(v)
data.mean <- mean(v)
data.var <- var(v)
data.dskew <- skewness(v)*data.var^1.5 # Denormalized skewness
data.dkurt <- (kurtosis(v) + 3)*data.var^2 # Denormalized kurtosis

```

Listing A.1: R code to generate production yield data and their moments for Example 1.

Only the first two moments and the ECDF information were considered in the DMPs. The weights for the moments were chosen such that the weight of the mean is the same as the weight of the variance, that is $w_{i,k} = 1.0/M_{i,k}^2$ and $w_{i,k} = 1.0/|M_{i,k}|$ for L^2 DMP and L^1 DMP, respectively. The weights for the deviations from the approximation to the ECDF are equal to one, $\omega_{i,j} = 1.0$. The ECDF data were estimated in MATLAB ([The MathWorks Inc., 2013](#)) with the function `ecdf` and a simplified GLF ($\beta_0 = 0$ and $\beta_1 = 1$) was fit to the ECDF data by using the function `lsqcurvefit` with the following options: 10,000 maximum function evaluations (`MaxFunEvals`), 10,000 maximum iterations (`MaxIter`), 10^{-12} function tolerance (`TolFun`), exact Jacobian. Initial guesses for the parameters β_2 , β_3 , and β_4 were 100, 10, and 1, respectively.

The parameters for the production planning LP model are given in the following tables.

Table A.1: First-stage production yield (θ_f at $t = 1$) [-]. For facilities $P2$ and $P3$, the yields remain the same at the second stage.

Facility	Group
$P1$	0.85
$P2$	0.7
$P3$	0.9

Table A.2: Product demand ($\xi_{m,t}$) [t].

Product	Time Period			
	1	2	3	4
C	20	10	15	17
D	15	7	12	10

Table A.3: Maximum inventory ($w_{m,t}^{\text{inv,max}}$) [t].

Product	Time Period			
	1	2	3	4
<i>D</i>	6	6	6	6

Table A.4: Maximum (minimum) capacity ($w_{m,t}^{\text{min}}$ and $w_{m,t}^{\text{max}}$) [t].

Facility	Time Period			
	1	2	3	4
<i>P1</i>	20 (5)	20 (5)	20 (5)	20 (5)
<i>P2</i>	18 (0)	18 (0)	18 (0)	18 (0)
<i>P3</i>	19 (2)	19 (2)	19 (2)	19 (2)

Table A.5: Selling price ($\text{SP}_{m,t}$) [\$/t].

Product	Time Period			
	1	2	3	4
<i>C</i>	1.75	1.75	1.71	1.71
<i>D</i>	1.10	1.10	1.18	1.18

Table A.6: Operating cost ($\text{OPC}_{f,t}$) [\$/t].

Facility	Time Period			
	1	2	3	4
<i>P1</i>	0.26	0.26	0.14	0.14
<i>P2</i>	0.16	1.16	0.13	0.13
<i>P3</i>	0.32	0.32	0.32	0.32

Table A.7: Purchase cost ($\text{PC}_{m,t}$) [\$/t].

Product	Time Period			
	1	2	3	4
<i>A</i>	0.05	0.05	0.05	0.05
<i>C</i>	1.00	1.00	1.00	1.00

Table A.8: Inventory cost ($\text{IC}_{m,t}$) [\$/t].

Product	Time Period			
	1	2	3	4
<i>D</i>	0.03	0.03	0.03	0.03

Table A.9: Raw material availability ($x_{A,t}^{\text{purch,max}}$) [t].

Raw Material	Time Period			
	1	2	3	4
A	30	25	27	23

The penalties for unmet demand ($\text{PEN}_{m,t}$), and maximum capacity and minimum capacity violations ($\text{PEN}_{f,t}$) are 10 times the selling prices, and proportional to the operating costs, respectively.

Appendix B Data for Example 2

Time series modeling and forecasting were implemented in the R programming language (R Core Team, 2013). For demonstration purposes, only ARIMA models (see Appendix ??) were fit to the time series data. The library used for fitting ARIMA models is called `forecast` (Hyndman *et al.*, 2013). In particular, the function `auto.arima` was used to automatically determine the best ARIMA model that fits the data according to the default information criterion (Akaike Information Criterion, AIC). Seasonality effects were allowed and no hold out sample was considered in the fitting process. The time series demand data for both products *C* and *D* were then modeled as ARIMA(1,0,0) processes with non-zero means. Future values were predicted with 95% level of confidence using the `predict` function in the `stats` library, which is part of R.

The constant variance of ARIMA models was estimated by the `sigma2` attribute of the ARIMA object. For the *LP Approach* in Subsection 3.2, the standard error of a forecast was estimated by setting the value of the argument `se.fit` of the `predict` function to `TRUE`. Covariance information was estimated through R's builtin function `ccf`, which estimates the cross-covariance function of two time series data at different lags. Since the *NLP Approach* involves alternating one-step-ahead forecasting with optimization, the estimated covariance corresponds to the lag 1 outcome of function `ccf`.

The parameters for the production planning LP model are the same as the data given in tables Tables A.3 to A.9. In addition, the production yield for all time periods are the same as those given in Table A.1. The product demand for every stage in each scenario tree is generated as described in Subsection 3.3.

References

- Abel, O., and Marquardt, W. 2000. Scenario-Integrated Modeling and Optimization of Dynamic Systems. *AIChE Journal*. 46(4):803–823.
- Abramowitz, M., and Stegun, I. A. 1965. *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. Dover Publications.
- Bartlett, R. 2013. *A Practitioner's Guide to Business Analytics: Using Data Analysis Tools to Improve Your Organization's Decision Making and Strategy*. The McGraw-Hill Companies.

- Bayraksan, G., and Morton, D. P. 2006. Assessing Solution Quality in Stochastic Programs. *Mathematical Programming*. 108(2-3):495–514.
- Becker, M., and Klößner, S. 2013. *PearsonDS: Pearson Distribution System*. R package version 0.95.
- Ben-Tal, A.; Ghaoui, L. E.; and Nemirovski, A. 2009. *Robust Optimization*. Princeton Series in Applied Mathematics.
- Beraldi, P.; De Simone, F.; and Violi, A. 2010. Generating Scenario Trees: A Parallel Integrated Simulation-Optimization Approach. *Journal of Computational and Applied Mathematics*. 233(9):2322–2331.
- Birge, J. R., and Louveaux, F. 2011. *Introduction to Stochastic Programming*. Springer, second edition.
- Chen, Y.; Adams II, T. A.; and Barton, P. I. 2011. Optimal Design and Operation of Flexible Energy Polygeneration Systems. *Industrial & Engineering Chemistry Research*. 50(8):4553–4566.
- Chiralaksanakul, A., and Morton, D. P. 2004. Assessing Policy Quality in Multi-stage Stochastic Programming. *Stochastic Programming E-Print Series*. (12):1–36. Retrieved from <http://edoc.hu-berlin.de/docviews/abstract.php?id=26770>.
- Colvin, M., and Maravelias, C. T. 2009. Scheduling of Testing Tasks and Resource Planning in New Product Development using Stochastic Programming. *Computers & Chemical Engineering*. 33(5):964–976.
- Dantzig, G. B. 1955. Linear Programming under Uncertainty. *Management Science*. 1(3-4):197–206.
- Davenport, T. H., and Harris, J. G. 2007. *Competing on Analytics: The New Science of Winning*. Harvard Business School Press.
- Dupačová, J.; Gröwe-Kuska, N.; and Römisch, W. 2003. Scenario Reduction in Stochastic Programming: An Approach using Probability Metrics. *Mathematical Programming: Series A*. 95:493–511.
- Eldar, Y. C., and Kutyniok, G. 2012. *Compressed Sensing: Theory and Applications*. Cambridge University Press.
- Feng, Y., and Ryan, S. M. 2013. Scenario Construction and Reduction Applied to Stochastic Power Generation Expansion Planning. *Computers & Operations Research* 40(1):9–23.
- Goel, V., and Grossmann, I. E. 2006. A Class of Stochastic Programs with Decision Dependent. *Mathematical Programming*. 108(2-3):355–394.
- Guillén, G.; Espuña, A.; and Puigjaner, L. 2006. Addressing the Scheduling of Chemical Supply Chains under Demand Uncertainty. *AIChE Journal*. 52(11):3864–3881.

- Gülpınar, N.; Rustem, B.; and Settergren, R. 2004. Simulation and Optimization Approaches to Scenario Tree Generation. *Journal of Economic Dynamics and Control*. 28(7):1291–1315.
- Gupta, V., and Grossmann, I. E. 2012. Modeling and Computational Strategies for Optimal Development Planning of Offshore Oilfields under Complex Fiscal Rules. *Industrial & Engineering Chemistry Research*. 51(44):14438–14460.
- Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Series in Statistics, second edition.
- Heitsch, H., and Römisch, W. 2009. Scenario Tree Reduction for Multistage Stochastic Programs. *Computational Management Science*. 6(2):117–133.
- Høyland, K., and Wallace, S. W. 2001. Generating Scenario Trees for Multistage Decision Problems. *Management Science*. 46(2):295–307.
- Høyland, K.; Kaut, M.; and Wallace, S. W. 2003. A Heuristic for Moment-Matching Scenario Generation. *Computational Optimization and Applications*. 24(2-3):169–185.
- Hyndman, R. J.; with contributions from; Athanasopoulos, G.; Razbash, S.; Schmidt, D.; Zhou, Z.; and Khan, Y. 2013. *forecast: Forecasting functions for time series and linear models*. R package version 4.06.
- Ji, X.; Zhu, S.; Wang, S.; and Zhang, S. 2005. A Stochastic Linear Goal Programming Approach to Multistage Portfolio Management Based on Scenario Generation via Linear Programming. *IIE Transactions*. 37(10):957–969.
- Kaut, M. 2003. *Scenario Tree Generation for Stochastic Programming: Cases from Finance*. Ph.D. Dissertation, Department of Mathematical Sciences. Norwegian University of Science and Technology.
- Kaut, M. 2013. A Copula-Based Heuristic for Scenario Generation. *Computational Management Science*. 25. DOI: 10.1007/s10287-013-0184-4.
- Kim, J.; Realff, M. J.; and Lee, J. H. 2011. Optimal Design and Global Sensitivity Analysis of Biomass Supply Chain Networks for Biofuels under Uncertainty. *Computers & Chemical Engineering*. 35(9):1738–1751.
- King, A. J., and Wallace, S. W. 2012. *Modeling with Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer Science+Business Media New York.
- Kleywegt, A. J.; Shapiro, A.; and Homem-de-Mello, T. 2001. The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM Journal of Optimization*. 12(2):479–502.
- Li, Z., and Ierapetritou, M. G. 2011. Capacity Expansion Planning Through Augmented Lagrangian Optimization and Scenario Decomposition. *AIChE Journal*. 58(3):871–883.

- Meyer, D.; Dimitriadou, E.; Hornik, K.; Weingessel, A.; and Leisch, F. 2012. *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6-1.
- Papoulis, A. 1991. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, Inc., third edition.
- Pinto-Varela, T.; Barbosa-Povoa, A. P. F. D.; and Novais, A. Q. 2009. Design and Scheduling of Periodic Multipurpose Batch Plants under Uncertainty. *Industrial & Engineering Chemistry Research*. 48(21):9655–9670.
- R Core Team. 2013. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. <http://www.R-project.org/>.
- Richards, F. J. 1959. A Flexible Growth Function for Empirical Use. *Journal of Experimental Botany*. 10(2):290–301.
- Roelofs, M., and Bisschop, J. 2013. *Advanced Interactive Multidimensional Modeling System (AIMMS)*. <http://www.aimms.com/>.
- Sahinidis, N. V. 2004. Optimization Under Uncertainty: State-of-the-Art and Opportunities. *Computers & Chemical Engineering*. 28(6-7):971–983.
- Shapiro, A. 2006. On Complexity of Multistage Stochastic Programs. *Operations Research Letters*. 34(1):1–8.
- Sundaramoorthy, A.; Evans, J. M. B.; and Barton, P. I. 2012. Capacity Planning under Clinical Trials Uncertainty in Continuous Pharmaceutical Manufacturing, 1: Mathematical Framework. *Industrial & Engineering Chemistry Research*. 51(42):13692–13702.
- Sutiene, K., and Pranevicius, H. 2007. Scenario Generation Employing Copulas. *Proceedings of the World Congress on Engineering. Volume II*. 777–784.
- The MathWorks Inc. 2013. *MATLAB® R2013a*. The MathWorks Inc., Natick, Massachusetts, United States. <http://www.mathworks.com/>.
- van der Vaart, A. W. 1998. *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press.
- Xu, D.; Chen, Z.; and Yang, L. 2012. Scenario Tree Generation Approaches using K-means and LP Moment Matching Methods. *Journal of Computational and Applied Mathematics*. 236(17):4561–4579.
- You, F.; Wassick, J. M.; and Grossmann, I. E. 2009. Risk Management for a Global Supply Chain Planning under Uncertainty: Models and Algorithms. *AIChE Journal*. 55(4):931–946.