Recent Advances in Mathematical Programming Techniques for the Optimization of Process Systems under Uncertainty

Ignacio E. Grossmann, Robert M. Apap, Bruno A. Calfa, Pablo Garcia-Herreros, Qi Zhang

Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, PA 15213, U.S.A.

Abstract

Optimization under uncertainty has been an active area of research for many years. However, its application in Process Systems Engineering has faced a number of important barriers that have prevented its effective application. Barriers include availability of information on the uncertainty of the data (ad-hoc or historical), determination of the nature of the uncertainties (exogenous vs. endogenous), selection of an appropriate strategy for hedging against uncertainty (robust/chance constrained optimization vs. stochastic programming), large computational expense (often orders of magnitude larger than deterministic models), and difficulty of interpretation of the results by non-expert users. In this paper, we describe recent advances that have addressed some of these barriers for mostly linear models.

Keywords: decision rule, robust optimization, stochastic programming, exogenous uncertainty, endogenous uncertainty, scenario generation.

1. Introduction

Optimization under uncertainty has been an active area of research in Process Systems Engineering. General reviews on various aspects of uncertainty can be found for instance in Sahinidis (2004), Li and Ierapetritou (2008) and Grossmann et al. (2014).

It is interesting to note that there is a long history of optimization under uncertainty in Process Systems Engineering. The initial work was mostly focused on optimal design problems using stochastic programming as the major framework. Examples include the papers by Takamatsu, Hashimoto, Shioya (1973), Dittmar and Hartmann (1976), Johns, Marketos, Rippin (1976) and Grossmann and Sargent (1978). The next phase addressed planning problems under uncertainty, where examples of papers in this area include Liu and Sahinidis (1996), Acevedo, Pistikopoulos (1998), Gupta and Maranas (2000), and Applequist, Pekny and Reklaitis (2000). The more recent trend has been to adopt robust optimization as a strategy for optimization under uncertainty, where it is interesting to point out that the work by Friedman and Reklaitis (1975) was well ahead of his time with linear programming with uncertain coefficients; other examples include the papers by Swaney and Grossmann (1985) with the emphasis on flexibility, and Lin, Janak and Floudas (2004) with applications in the scheduling area. Finally, the other major approach has been based on chance constrained optimization that has been pioneered by Prof. Wozny (e.g. Arellano-Garcia, Martini, Wendt, Li and Wozny (2003)).

Optimization under uncetainity has been motivated by the fact that parameters involved in optimization models for synthesis, design, planning, scheduling and supply chain are often uncertain. Typical parameters include product demands, prices of chemicals, product yields, oil reservoir sizes, and technical parameters like kinetic constants or transfer coefficients. Yet, despite the importance of accounting for uncertainties, the impact of techniques for optimization under uncertainty at the level of applications has been limited. This has been due to a number of modeling and computational barriers. In this paper, we first provide a brief overview of the area and then describe some recent advances that are facilitating the wider deployment and application of optimization under uncertainty.

A major modeling decision in optimization under uncertainty is whether one should rely on robust optimization (Ben-Tal et al., 2009; Lin et al., 2004; Li and Ierapetritou, 2008b), or whether one should use stochastic programming (Ierapetritou and Pistikopoulos, 1994; Subrahmanyam et al., 1994; Iyer and Grossmann, 1998; Schultz, 2003; Ahmed and Garcia, 2003; Li and Ierapetritou, 2012). The basic idea in the robust optimization approach is to guarantee feasibility over a specified uncertainty set, while in stochastic programming a subset of decisions are made by anticipating that recourse actions can be taken once the uncertainties are revealed over a pre-specified scenario tree with discrete probabilities of the uncertainties. In general this means that robust optimization will predict more conservative results compared to stochastic programming.

On the one hand, the robust optimization approach tends to be more appropriate for short-term scheduling problems in which feasibility over a specified set of uncertain parameters is a major concern, and when there is not much scope for recourse decisions. The stochastic programming approach, on the other hand, tends to be more appropriate for long-term production planning and strategic design decisions, since it does not fix all the decisions at the initial point of the planning horizon as it allows recourse decisions in future times to adapt in response to how the uncertainties are revealed. It should also be noted that chance constrained optimization can be regarded as a generalization of robust optimization in which distribution functions are specified for the uncertainties with which a level of probability for meeting the constraints feasibility is specified (Li et al., 2008).

Computationally, stochastic programming problems tend to be very expensive to solve. This is particularly true if one has to deal with both exogenous and endogenous parameters, where the realization of the latter depends on the decisions taken (e.g., oilfield size), while the former are independent of decisions (e.g., demands) (Jonsbråten, 1998). Chance constrained programming is also computationally challenging, especially when the level of probability is specified for the joint constraints. Robust optimization tends to be the least computationally expensive, especially for linear problems. In all cases, however, it is of great importance to have as a basis a computationally efficient deterministic model since adding uncertainty unavoidably adds complexity to the computations. Aside from the computational issues, there is also the question of how to specify the uncertainties (e.g., an intuitive guess or through statistical analysis of historical data) and how to interpret the results that are predicted from the various models.

In this paper, we address the following specific challenges in optimization under uncertainty: (a) how to account for recourse in robust optimization to make it less conservative, (b) how to effectively solve two- or multi-stage stochastic optimization problems to avoid excessive computational times, (c) how to effectively handle both exogenous and endogenous uncertainties in multi-stage stochastic programming, and (d) how to incorporate historical data in the generation of scenarios instead of relying on intuition. As will be shown, significant progress has been made to address these issues in MILP models with uncertainty.

2. Decision Rule Approach for Modeling Recourse in Robust Optimization

Robust optimization (Ben-Tal et al., 2009) is one approach for incorporating uncertainty in optimization models. The uncertainty is specified in terms of an uncertainty set in which any point is a possible realization of the uncertainty. The goal is to find a solution that is feasible for all possible realizations of the uncertainty while minimizing (or maximizing) the objective function. Since the worst-case scenario is one of the possible realizations, a robust optimization model returns a solution that is optimal for this particular scenario. However, considering the worst case is often overly conservative. One can reduce the level of conservatism by appropriately adjusting the size of the uncertainty set. Bertsimas and Sim (2004) propose such an approach in which a pre-specified "budget" parameter limits the number of uncertain parameters that can change at the same time.

The general robust formulation for a linear model is as follows:

$$\min_x \{c^T x : A(u)x \le b \quad \forall\, u \in U\} \tag{1}$$

where the parameter u is uncertain and defined over the corresponding uncertainty set U in which we assume the uncertainty budget is included. Eq. (1) corresponds to a semi-infinite programming problem, but can be simplified by considering the dual of each row in the matrix A (robust counterpart), which yields a finite dimensional optimization problem. A drawback with this approach is that specifying the uncertainty set U is not trivial, not only because the uncertainty set might not be well known, but also because the user has to specify the uncertainty "budget" ranging from very conservative (all parameters vary independently) to less risk-averse (e.g., limiting the number of independent variations). Another reason for over-conservatism in traditional robust optimization is the disregard of recourse (i.e., reactive actions after the realization of the uncertainty), which is a very unrealistic assumption in many cases, such as in problems involving investment and long-term contract decisions. In this section, we present a recent development in robust optimization that allows recourse to a certain extent, and demonstrate the effectiveness of this method by applying it to an industrial scheduling problem.

2.1. The Affine Decision Rule Approach

Consider the following multi-stage (T stages) optimization problem under uncertainty:

$$\min_{x} \left\{ c^T x_1 : A_1(u) x_1 + \sum_{t=2}^{T} A_t x_t(u) \leq b \quad \forall\, u \in U \right\} \tag{2}$$

where x_t is the vector of the t-th-stage variables. We assume that the objective function only depends on x_1 and that only matrix A_1 is uncertain (fixed recourse). While x_1 does not depend on u, x_t for $t \geq 2$ are recourse variables and depend on the realization of the uncertainty. Note that x_t only depends on the u that are realized up to stage t.

The problem given in Eq. (2) cannot be solved as such since the set of possible functions for $x_t(u)$ is infinitely large. The idea in the decision rule approach (Ben-Tal et al., 2004), also referred to as adjustable robust optimization, is to restrict oneself to a certain type of functions for $x_t(u)$, in particular to the set of affine functions. Hence, we set $x_t(u) = \alpha_t + \mathrm{B}_t u$, and we obtain the following robust formulation by constraint-wise construction:

$$\min_{x_1, \alpha, B} \left\{ c^T x_1 : \max_{u \in U} \left\{ a_{1,i}^T(u) x_1 + \sum_{t=2}^{T} a_{ti}^T (\alpha_t + \mathrm{B}_t u) \right\} \leq b_i \quad \forall\, i \right\} \tag{3}$$

which can be reformulated into a single-level problem for certain types of uncertainty sets by using techniques applying strong duality. This results in a robust counterpart formulation in which the decision variables are x_1 as well as the parameters for the affine decision rules, α and B. By applying the decision rule approach, the multi-stage problem is transformed into a single-stage problem to which the classic robust optimization reformulation is applied. Obviously, it is quite restrictive to only consider affine functions; however, this allows us to retain computational tractability and still account for recourse to a certain extent (for rigorous treatment of recourse see Grossmann et al. (2014)).

The decision rule approach has been successfully applied to various operations research problems, such as inventory management (Ben-Tal et al., 2004), project management (Chen and Zhang, 2009), and logistics planning (Ben-Tal et al., 2011). In fact, it cannot only be applied to robust optimization, but also to stochastic programming (Kuhn et al., 2011). However, application of this approach in the process systems engineering literature is still very scarce.

2.2. Industrial Case Study: Air Separation Plant Providing Interruptible Load

To ensure the stability of the power grid, backup capacities are called upon when electricity supply does not meet demand due to unexpected changes in the grid. Traditionally, such operating reserve is provided by generating facilities with short ramp-up times. Alternatively, as part of the demand response efforts in recent years, also large electricity consumers are encouraged to provide operating reserve in the form of load reduction capacities referred to as interruptible load. Selling interruptible load is financially attractive because the provider is rewarded even when no actual load reduction is required. However, the challenge lies in the uncertainty that one does not know in advance when load reduction will be requested; noncompliance would result in extremely high penalties or one may not even be allowed to participate in the market. Hence, a robust solution is required, while recourse actions need to be considered in order to realistically model the decision-making process.

Zhang et al. (2015b) have developed an MILP discrete-time scheduling model for power-intensive continuous industrial processes providing interruptible load, which applies the decision rule approach to model recourse in a multistage decision-making setting. In the proposed model, the here-and-now decisions are the ones related to the base operation (no load reduction required during the entire scheduling horizon) of the plant as well as how much interruptible load to provide. The wait-and-see decisions are production and purchase deviations from the base operation. To adjust the extent of recourse, the parameter $\bar{\zeta}$ is introduced, denoting the number of uncertain parameters from previous time periods that are incorporated in the linear decision rules. If $\bar{\zeta} = 0$, only the uncertain parameter from the current time period appears in the decision rules; hence, the only possible recourse actions are the reduction in production rate when load reduction is requested and additional product purchase in the same time periods. If $\bar{\zeta} > 0$, i.e. uncertain parameters from previous time periods are also taken into account, lost production can also be made up by increasing production or purchase in time periods after the load reduction occurred. Furthermore, in order to reduce the level of conservatism, the budget uncertainty set proposed by Zhang et al. (2015a) is applied, which sets a limit to the number of time periods in which maximum reserve dispatch can occur.

Applying the proposed model to an industrial air separation case study, for which the data are provided by Praxair, the scheduling problem is solved for a one-week time horizon with an hourly time discretization. The uncertainty set is chosen such that request for maximum load reduction can happen up to 7 times a week, which is a fairly conservative assumption. In practice, the budget parameters can be chosen based on historical data; alternatively, depending on the market, there may be a strict limit on the number of times in which load reduction can be requested during a specific time horizon, which can be used to set the budgets.

The solution strongly depends on the extent of recourse that is considered in the model. With $\bar{\zeta} = 0$, a cost reduction of 1.2% is achieved compared to the case in which no interruptible load is provided. These cost savings further increase by more than 50% if $\bar{\zeta}$ is changed to 23. However, this improvement in the quality of the solution comes at the cost of deteriorating computational performance. In the case of $\bar{\zeta} = 0$, the model has 3,282 binary variables, 82,670 continuous variables, and 84,604 constraints, while for $\bar{\zeta} = 23$, the number of binary variables remains the same, but the numbers of continuous variables and constraints increase to 330,242 and 325,000, respectively. The computation times required to solve the models to 0.1% optimality gap using CPLEX 12.6 on an Intel Core i7-2600 machine at 3.40 GHz with 8 processors and 8 GB RAM are 185 s and 6,476 s. The parameter $\bar{\zeta}$ can be further increased (up to 167). However, computational experiments show only marginal improvement in the solution for $\bar{\zeta} > 23$; hence $\bar{\zeta} = 23$ is chosen as a good trade-off between level of conservatism and problem size.

The results for the case with greater extent of recourse, i.e. $\bar{\zeta} = 23$, are shown in Fig. 1 and Fig. 2. Along with the electricity and interruptible load prices, Fig. 1 shows the target load profile for the plant as well as the amount of interruptible load provided, which obviously has to be less than the target electricity consumption. For liquid oxygen (LO2), one of the products, Fig. 2 shows the inventory profile and the corresponding product flows as well as the cumulative recourse actions in terms of changes in production and purchase rates. Negative production recourse indicates time periods in which interruptible load is provided. One can see that the vast majority of the lost production is made up by increasing production after load reduction (positive production recourse).

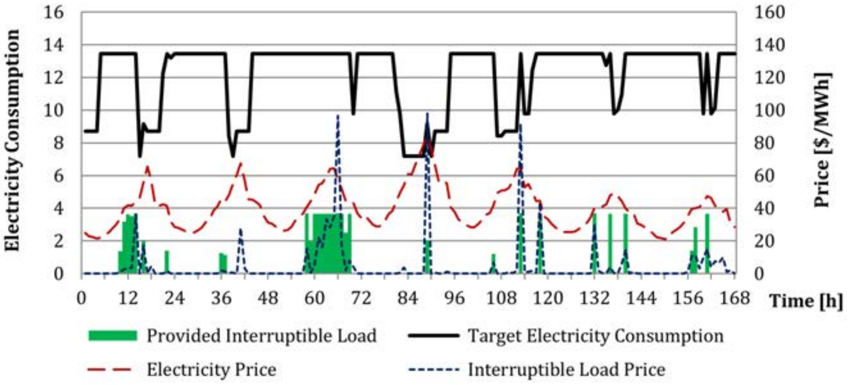


Fig. 1. Target electricity consumption profile and provided interruptible load for the case of $\bar{\zeta} = 23$, and price profiles.

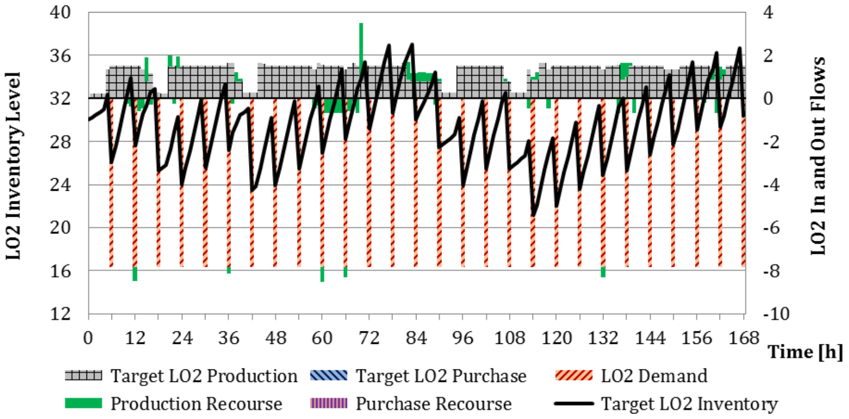


Fig. 2. Target and recourse LO2 flows and target inventory profile.

2.3. Other Applications of Robust Optimization in Process Systems Engineering

In the area of process systems engineering, robust optimization (mostly the "traditional" approach without recourse) has mainly been applied to production scheduling problems (Li and Ierapetritou, 2008a,b). Lin et al. (2004) consider uncertainty in processing times, product demands, and prices in a continuous-time scheduling model, where the uncertain parameters can take any values between given lower and upper bounds. This work is extended by Janak et al. (2007) who consider uncertainty with known probability distribution. A similar approach is taken by Verderame and Floudas (2009) to address demand due date and amount uncertainty in batch scheduling. Mitra et al. (2012) apply robust optimization to model uncertainty electricity prices. The proposed approach features different possible price ranges and can account for correlated data. Vujanic et al. (2012) find robust solutions to batch scheduling problems in which the start times of scheduled tasks are uncertain, which may be caused by load shifting required to meet operating reserve demand. Uncertainty in operating reserve demand has also been considered by Zhang et al. (2015). Amaran et al. (2015) combine robust optimization and stochastic programming to address the problem of medium-term maintenance turnaround planning for integrated chemical sites where the uncertainty lies in the turnaround duration. In one of the few applications of robust optimization to logistics problems, Gounaris et al. (2013) derive robust counterparts for the capacitated vehicle routing problem.

Li et al. (2011) apply robust optimization to linear and mixed-integer linear optimization problems, and present a systematic study of the robust counterparts for various uncertainty sets and their geometric relationships. In a subsequent work, Li et al. (2012) derive probabilistic guarantees on constraint satisfaction for these robust counterparts.

The design of robust or flexible chemical processes under uncertainty has been dominated by the concept of flexibility analysis, which was first proposed and then further developed by Grossmann and coworkers (Halemane and Grossmann, 1983; Swaney and Grossmann, 1985; Grossmann and Floudas, 1987). However, there is a strong connection between flexibility analysis and robust optimization; recent insights on the relation between these two approaches are presented in Zhang et al. (2015).

3. Two-Stage and Multistage Stochastic Programming

Stochastic programming is the framework that models mathematical programs with uncertainty by optimizing the expected value over the possible realizations (Birge and Louveaux, 2011). In general, the expected value is computed by integrating over the set of uncertain parameters, which might be a challenging task. In the case of discrete uncertainty sets with finite support, the realizations can be characterized with a finite number of scenarios, simplifying the calculation of the expected value. Accordingly, stochastic programming is often regarded as a scenario-based approach for optimization under uncertainty. The formulations can accommodate decision making at different stages according to the sequence in which uncertainty reveals. The stages imply a discrete time representation of the problem and establish the information of the uncertain parameters available at that time. The potential paths in which discrete uncertain parameters might evolve are represented in a scenario tree as shown in Fig. 3. In these trees, each node is a decision-making instance with known realization of the uncertain parameters up to the current state; potential future realizations are represented with branches from the given node.

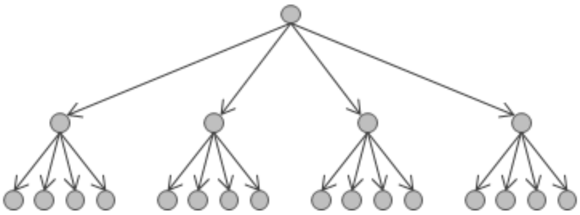


Fig. 3. Tree representation of scenarios in a stochastic program with three stages.

The simplest stochastic programming formulation only considers decisions that are made before uncertainty reveals. It is called single-stage stochastic programming or stochastic programming without recourse. Among the stochastic programs that consider recourse, the most widely used formulation is the Mixed-Integer Linear Program (MILP) with continuous recourse in a second stage. The two-stage stochastic programming formulation divides the decisions into two sets: here-and-now decisions that are made before uncertainty reveals, and wait-and-see decisions that are independent for each scenario. The typical formulation of a linear two-stage stochastic programming problem is presented in Eq. (4),

$$\begin{aligned} &\min_{x_1 \in X_1} \quad c^T x_1 + \mathbb{E}_{s\in S}[Q(x_1)] \\ &\text{s.t.}\ \ Q(x_1) = \min_{x_2^s \in X_2^s}\ d_s^T x_2^s \\ &\qquad\qquad\quad \text{s.t.}\ \ W x_2^s \le h_s - T_s x_1 \end{aligned} \tag{4}$$

where x_1 is the vector of first-stage decisions in mixed-integer polyhedral set X_1, x_2^s is the vector of second-stage (recourse) decisions in polyhedral set X_2^s, and s is the index for scenarios. An important property of the formulation presented in Eq. (4) is that the feasible region for the first-stage variables x is a convex polyhedron (Birge and Louveaux, 2011). Based on this property and assuming that the second-stage problems ($Q(x,s)$) are bounded, an MILP model that explicitly calculates the expected value can be derived by including all second-stage problems in the formulation. This reformulation, known as the deterministic equivalent of the stochastic programming problem, is presented in Eq. (5).

$$
\begin{aligned}
\min \quad & c^T x_1 + \sum_{s \in S} p_s (d_s^T x_2^s) & & \\
\text{s.t.} \quad & W x_2^s \le h_s - T_s x_1 & \forall\, s \in S & \qquad (5) \\
& x_1 \in X_1;\ \ x_2^s \in X_2^s & &
\end{aligned}
$$

The benefits of using a stochastic programming model can be quantified by the Value of the Stochastic Solution (VSS). The VSS is the difference between the expected values obtained from implementing the solution yielded by the stochastic formulation and the solution yielded by a deterministic formulation that substitutes the uncertain parameters with their nominal values. The expected value of the deterministic formulation is calculated by solving the problem, implementing the here-and-now decisions, and evaluating the scenarios with their optimal recourse.

The model in Eq. (4) can also be extended to a multistage stochastic programming model. The tree corresponding to a three-stage problem has the structure shown in Fig. 3. The solution of multistage stochastic programming problems is considerably harder, and special care must be taken to avoid anticipating the uncertain parameters that have not been revealed. The general formulation of a three-stage stochastic programming problem is presented in Eq. (6),

$$
\begin{aligned}
& \min_{x_1 \in X_1} \quad c^T x_1 + \mathbb{E}_{s_2}[Q_2(x_1)] \\
& \text{s.t.} \quad Q_2(x_1) = \min_{x_2^s \in X_2^s} \quad d_s^T x_2^s + \mathbb{E}_{s_3|s_2}[Q_3(x_2^s)] \qquad\qquad (6) \\
& \qquad\qquad\qquad \text{s.t.}\ W_s x_2^s \le h_s - T_s x_1 \\
& \qquad\qquad\qquad\qquad Q_3(x_2^s) = \min_{x_3^s \in X_3^s} \quad f_s^T x_3^s \\
& \qquad\qquad\qquad\qquad\qquad \text{s.t.} \quad V_s x_3^s \le g_s - U_s x_1 - W_s x_2^s
\end{aligned}
$$

One way of transforming the formulation in Eq. (6) into its deterministic equivalent is to generate a set of copied variables for each path from the root node to the branches, and introduce non-anticipativity constraints that force copied variables to have the same values according to uncertainty revealed at each stage (Ruszczyński, 1997). This *alternative representation*, shown in Fig. 4, has the advantage of being relatively easy to implement. Solution methods for large stochastic programming problems is still an active area of research due to the large number of scenarios needed for industrial applications and the rapid growth of scenarios in multistage stochastic programming. The development of decomposition methods has allowed addressing large-scale stochastic programming problems.

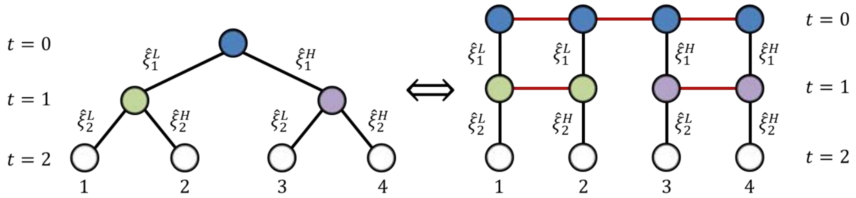


Fig. 4. An exogenous scenario tree and its alternative representation.[1]

3.1. Decomposition Methods for Stochastic Programming Problems

Methods used to solve large stochastic programming problems leverage the scenario structure that produces a block-diagonal shape in the constraints. Similarly to the application of decomposition methods to other types of mathematical programs, the strategies used for stochastic programming find the optimal solution by iteratively solving problems of low complexity. The scenario structure of stochastic programs implies that most of the variables and constraints in the formulation have no direct interaction with each other. In particular, it is easy to identify a relatively small set of complicating variables and complicating constraints connecting the scenarios. This observation is gives rise to the main decomposition methods for stochastic programming: Lagrangean decomposition and L-shaped method. Other decomposition strategies include Progressive Hedging and Nested Decomposition procedures.

3.1.1. Lagrangean Decomposition

Given that a small set of constraints is responsible for linking decisions across scenarios in stochastic programs, relaxing these complicating constraints allows finding the optimal solution for each scenario independently. The Lagrangean decomposition method (Guignard and Kim, 1987) uses Lagrangean relaxation over the *alternative representation* of a stochastic program to dualize the non-anticipativity constraints (see Fig. 4).

An implementation of this dual decomposition method to stochastic programs was presented by Carøe and Schultz (1999). The solution of the relaxed subproblems yields a lower bound for the minimization of the original problem. This relaxation is improved iteratively by updating the Lagrange multipliers associated with the non-anticipativity constraints; the multipliers can be updated using different strategies, including subgradient and cutting plane methods. The main difficulty with the application of Lagrangean decomposition is that the solution of the Lagrangean relaxation is likely to violate the relaxed constraints and it might be difficult to find feasible solutions to the original problem. Additionally, the optimal solution of the Lagrangean dual only provides a lower bound to the original problem and

[1] From Apap and Grossmann (2015).

the duality gap might not be satisfactory to assess the quality of the optimal solution. Therefore, Lagrangean decomposition is often implemented in a branch-and-bound algorithm that closes the duality gap and with a heuristic that provides feasible solutions. The main advantage of Lagrangean decomposition is that it can be used to solve two-stage and multistage stochastic programming problems with mixed-integer variables in all stages.

3.1.2. L-shaped Method

The L-shaped method is the implementation of Benders decomposition to linear two-stage stochastic programming problems with continuous recourse (Van Slyke and Wets, 1969), which means that integer variables can only be handled in the first stage. The method considers first-stage variables as complicating because the subproblem corresponding to each scenario can be solved independently once they are fixed. Every iteration of the L-shaped method solves the scenario subproblems to evaluate a first-stage solution and a relaxed master problem that generates new candidate solutions for the first-stage variables. The optimal primal and dual solutions of the subproblems provide an upper bound on the original problem, a candidate solution (if feasible), and dual information that can be used to generate Benders cuts approximating the feasible region of the original problem in the space of the first-stage variables. These cutting planes are included iteratively in the master problem, improving the description of the feasible region and the quality of the first-stage solutions that it produces. The algorithm is guaranteed to converge to the optimal solution of the original problem in a finite number of iterations. The rate of convergence of the L-shaped method strongly depends on the linear programming relaxation of the problem; in subproblems with multiple optima, judicious selection of the alternative Benders cuts can greatly improve the convergence of the algorithm (Magnanti and Wong, 1981). Additionally, the scenario structure of the stochastic program can be exploited in the L-shaped method to generate independent cuts for each subproblem in every iteration. The addition of multi-cuts to the master problem often produces a major reduction in the number of iterations required for convergence (Birge and Louveaux, 1988).

3.2. Example of Two-Stage Stochastic Programming

The design of a resilient supply chain with risk of disruptions is presented by García-Herreros et al. (2014). The problem is formulated as a two-stage stochastic program. In the first stage, distribution centers (DCs) are selected from a set of candidate locations and their capacities are established. In the second stage, the demand assignment decisions are made according to the selected DCs and random disruptions in each scenario. The example includes: 1 production plant, 9 candidate locations for DCs, and 30 customers with demands for 2 commodities. The parameters of the instance were generated randomly. The candidate DCs have disruption probabilities between 2% and 10%. The number of scenarios in the full-space problem is $2^9 = 512$. The design is based on a time-horizon (N) of 365 days; on this time-scale, investment cost can be interpreted as annualized cost.

The instance is used to illustrate the use of the L-shaped method, the benefits of strengthening the master problem, and the impact of solving a reduced subset of relevant scenarios. The selected relevant subset of scenarios includes scenarios with up to 4 simultaneous disruptions, for a total of 256 scenarios with probability equal to 99.99%. The full-space model yields a total cost of $7,225,447, while the reduced model yields a similar cost of $7,224,591. Both solutions predict the same design decisions and therefore the same investment cost. The largest difference in the results is in the expected cost of penalties. Both problems were solved to 0% optimality tolerance using the special L-shaped algorithm described in García-Herreros et al. (2014). Both formulations involve 9 binary variables; the full-space model has 318,479 constraints and 309,263 continuous variables, while the reduced-space model has 159,247 and 154,639, respectively. The solution times for the multi-cut algorithm were 281 s and 151 s, respectively, while the Strengthened multi-cut algorithm required 176 s and 89 s, respectively. As can be seen in Fig. 5, the use of the strengthened multi-cut master problem reduces the solution time because it requires fewer iterations. The solution times for the full-space and the reduced instances without any decomposition strategy, using GUROBI 5.5.0, were 3,349 s and 1,684 s, respectively. The much smaller solution times obtained using the decomposition algorithm demonstrate that the proposed methodology is effective to solve large-scale instances of high computational complexity.

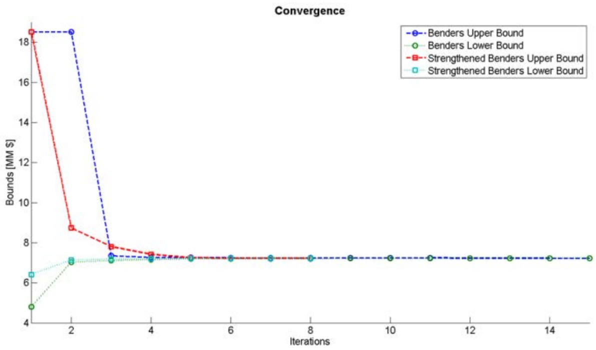


Fig. 5 Convergence of Benders algorithms for the full-space instance of the large-scale example.

3.3. Other Applications of Stochastic Programming in Process Systems Engineering

Stochastic programming has found extensive applications in process synthesis. Halemane and Grossmann (1983) presented a two-stage approach for the design of flexible processes under uncertainty. The design problem is formulated to minimize a cost function and to guarantee feasible operation over a polyhedral uncertainty set. The design and operation of multiproduct batch plants under uncertain demand was modeled as a nonconvex two-stage stochastic program by Ierapetritou and Pistikopoulos (1995); they exploit the special properties of the problem and use global optimization techniques to find the design with minimum investment and operating cost. Cheng et al. (2003) developed a multi-criteria formulation for design and planning under uncertainty that models technology availability as a Markov chain and demands with discrete probability distributions. The multi-period decision-making process is solved using a dynamic programming algorithm in which the single-period problems are two-stage stochastic programs. Pintarič and Kravanja (2004) proposed a recursive two-stage strategy for the synthesis of flexible processes under uncertainty whose computational complexity depends on the number of design variables and not on the number of stochastic parameters. In the first stage, a potential structure and size of the processes are determined; in the second-stage, additions to the model are identified to improve the operability of the system. They use the Central Basic Point to approximate the expected value of the objective function.

Supply chain design and management has been another fruitful area for stochastic programming. Tsiakis et al. (2001) developed a two-stage stochastic programming formulation for the design of multi-echelon supply chains under demand uncertainty. You et al. (2009) applied the L-shaped method to solve a supply chain planning problem under uncertainty considering risk management; they implemented the planning strategy in a rolling horizon to evaluate the potential savings of the stochastic programming formulation. The design and planning of closed-loop supply chains with stochastic demand and recovery of final products was presented by Baptista et al. (2012). The proposed two-stage stochastic programming model is solved using the L-shaped method to expedite the solution of large-scale problems. Jung et al. (2004) proposed a multistage stochastic programming formulation for multi-period supply chain planning; they developed a solution method that iterates between a rolling horizon simulation evaluating the schedules obtained from a deterministic problem and an outer loop that improves the safety stock targets using a gradient-based search. The solution of scheduling problems in a rolling horizon has also been addressed by Sand and Engel (2004) using two-stage stochastic programming with mixed-integer recourse. Their proposed solution approach uses aggregated master problems and detailed scheduling problems to optimize an expected value function including a measure of risk.

4. Multistage Stochastic Programming under Endogenous and Exogenous Uncertainties

Uncertain parameters can be classified into two categories depending upon the way the uncertainty is resolved: *exogenous*, where realizations occur independently of process decisions; and *endogenous*, where the realizations are influenced by the decisions (Jonsbråten, 1998). In the endogenous case, the decisions may influence the realizations in at least two ways: by altering the underlying probability distribution for the uncertain parameters, or by affecting the time at which we observe the realizations (Goel and Grossmann, 2006). Here, we will consider only the second class of endogenous uncertainty, where decisions affect the timing of realizations.

In the context of process systems engineering, exogenous uncertainties often correspond to market uncertainties, such as oil prices. This is the type that is most commonly considered in the literature. Reviews of this area can be found in Birge (1997), Schultz (2003), and Sahinidis (2004). Endogenous uncertainties, on the other hand, are technical uncertainties, such as oilfield size, that are only resolved after a decision is made at a given point in time (e.g., to drill a particular oilfield). This type belongs to a much newer area of research and has received far less attention in the literature. In fact, the first publication in this area was introduced by Jonsbråten et al. (1998) less than 20 years ago. Selected references in process systems engineering include applications in oil- and gas-field planning (Goel and Grossmann, 2004; Tarhan et al., 2009; Gupta and Grossmann, 2014), process network planning (Goel and Grossmann, 2006; Tarhan and Grossmann, 2008; Gupta and Grossmann, 2011), planning and scheduling in the pharmaceutical industry (Colvin and Maravelias, 2008; Colvin and Maravelias, 2010; Christian and Cremaschi, 2014), and open pit mine production scheduling (Boland et al., 2008).

Surprisingly, although many problems contain both endogenous and exogenous uncertainties (e.g., uncertain field sizes and uncertain oil prices), optimization under both types has been largely unexplored in the literature. As far as we are aware, Goel and Grossmann (2006) is the only existing comprehensive survey on multistage stochastic programming problems of this class.

In the case where the uncertainty is purely exogenous, the shape of the scenario tree is known in advance, since exogenous realizations occur automatically in each time period. This was shown previously in Fig. 4 with the standard form of the tree, as well as its alternative representation which gives each scenario a unique set of nodes.

In the case of endogenous uncertainty, however, the shape of the scenario tree is conditional, since the timing of realizations depends on the process decisions. As shown in Fig. 6, we use a superstructure form of the alternative tree in order to capture all possible outcomes (Apap and Grossmann, 2015).

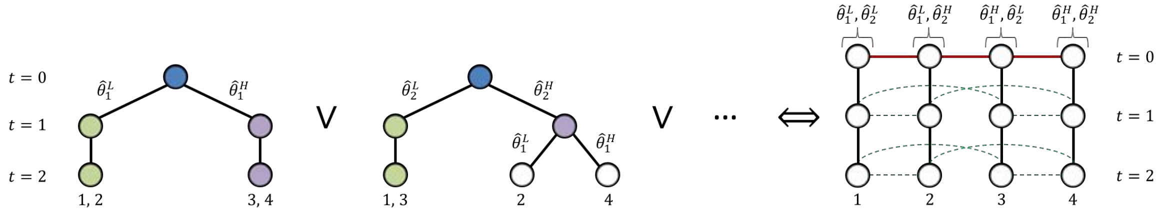


Fig. 6. A superstructure representation for endogenous scenario trees.

For simplicity, we assume that the scenarios correspond to all possible combinations of realizations of the uncertain parameters. In other words, the set of scenarios in the exogenous scenario tree corresponds to a Cartesian product over the sets of realizations for the exogenous parameters (denoted by R_X), and, similarly, the set of scenarios in the endogenous scenario tree corresponds to a Cartesian product over the sets of realizations for the endogenous parameters (denoted by R_N). For the case of both endogenous and exogenous parameters, we generate the set of scenarios by simply taking the Cartesian product of all possible combinations of realizations of the endogenous parameters and all possible combinations of realizations of the exogenous parameters, $R_N \times R_X$. This is equivalent to copying the exogenous scenario tree for each possible combination of realizations of the endogenous parameters. We then link these exogenous trees by adding first-period and endogenous non-anticipativity constraints, thereby producing what we refer to as a 'composite' scenario tree, as shown in Fig. 7.

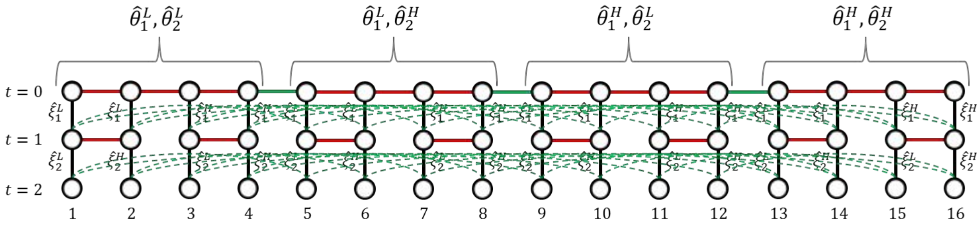

$\hat{\theta}_1^L, \hat{\theta}_2^L$
$\hat{\theta}_1^L, \hat{\theta}_2^H$
$\hat{\theta}_1^H, \hat{\theta}_2^L$
$\hat{\theta}_1^H, \hat{\theta}_2^H$
$t = 0$
$t = 1$
$t = 2$
$\hat{\xi}_1^L$
$\hat{\xi}_1^H$
$\hat{\xi}_2^L$
$\hat{\xi}_2^H$
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Fig. 7. A 'composite' scenario tree for endogenous and exogenous realizations.

The general model for this class of problems, originally proposed by Goel and Grossmann (2006), is a hybrid mixed-integer linear disjunctive program. A simplified, compact form of the updated model (Apap and Grossmann, 2015) is given in Eq. (7) – (15).

$$\min_{y} \phi = \sum_{s \in S} p^s \sum_{t \in T} c_t^s y_t^s \qquad (7)$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} A_{\tau,t}^s y_\tau^s \leq a_t^s \qquad \forall\, t \in T,\ \ s \in S \qquad (8)$$

$$y_1^s = y_1^{s'} \qquad \forall\, (s, s') \in SP_F \qquad (9)$$

$$y_{t+1}^s = y_{t+1}^{s'} \qquad \forall\, (t, s, s') \in SP_X \qquad (10)$$

$$y_{t+1}^s = y_{t+1}^{s'} \qquad \forall\, (t, s, s') \in SP_N,\ \ t \in T_E \qquad (11)$$

$$\begin{bmatrix} Z_t^{s,s'} \\ y_{t+1}^s = y_{t+1}^{s'},\ \ t < |T| \end{bmatrix} \vee \left[\neg Z_t^{s,s'} \right] \qquad \forall\, (t, s, s') \in SP_N,\ \ t \in T_C \qquad (12)$$

$$Z_t^{s,s'} \Leftrightarrow F(y_1^s, y_2^s, \dots, y_t^s) \qquad \forall\, (t, s, s') \in SP_N,\ \ t \in T_C \qquad (13)$$

$$y_t^s \in Y_t^s \qquad \forall\, t \in T,\ \ s \in S \qquad (14)$$

$$Z_t^{s,s'} \in \{True, False\} \qquad \forall\, (t, s, s') \in SP_N,\ \ t \in T_C \qquad (15)$$

The objective function, Eq. (7), minimizes the total expected cost associated with decisions y_t^s, weighted by the probability of each scenario, p^s. Eq. (8) represents constraints that govern decisions y_t^s and link decisions across time periods. First-period non-anticipativity constraints (NACs) are given by Eq. (9), exogenous NACs are given by Eq. (10), Eq. (11) gives endogenous NACs for the initial time periods before the endogenous uncertainty can be resolved, and disjunctive constraint (12) conditionally enforces endogenous NACs for the remaining time periods based on the value of Boolean variable $Z_t^{s,s'}$. The value of $Z_t^{s,s'}$ is determined by an uncertainty resolution rule in Eq. (13); its respective domain is given in Eq. (15). Bounds and integrality restrictions on decisions y_t^s are specified by mixed-integer set Y_t^s in Eq. (14).

Because the number of NACs grows exponentially as the number of time periods, uncertain parameters, or realizations increases, most problems of practical interest are too large to be solved directly with commercial MILP solvers. In order to eliminate redundant NACs, a number of theoretical properties have been proposed by Goel and Grossmann (2006), Gupta and Grossmann (2011), and Apap and Grossmann (2015) based on the concepts of symmetry, adjacency, transitivity, and scenario grouping. We apply these properties to the preceding model through the definitions of scenario-pair sets SP_F, SP_X, and SP_N (see Apap and Grossmann (2015) for further details). Although these properties drastically reduce the number of constraints, the resulting model is often still intractable and special solution approaches are required.

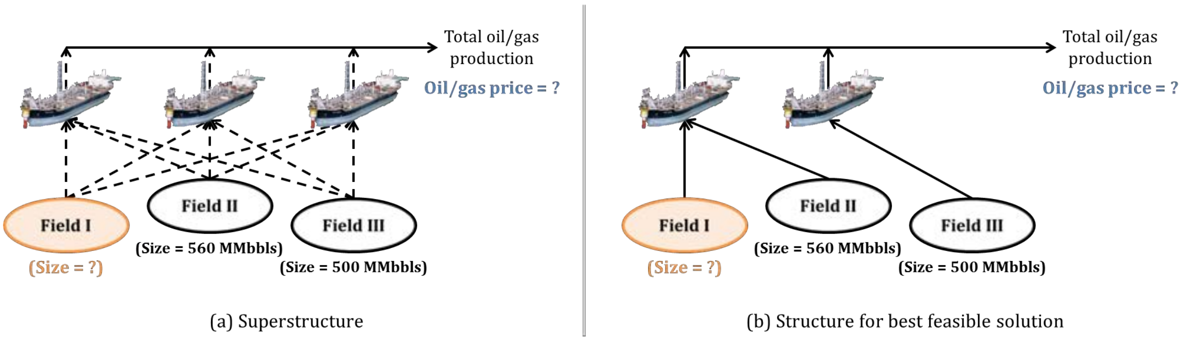
One effective approach is Lagrangean decomposition (LD), in which the complicating non-anticipativity constraints are dualized in order to decompose the problem into independent scenario subproblems that can be solved in parallel. This was previously discussed in section 3.1.1. Goel and Grossmann (2006) also developed a Lagrangean-duality-based branch-and-bound algorithm for this class of problems. More recently, a sequential scenario decomposition heuristic (SSD) has been proposed that involves sequentially solving endogenous MILP subproblems to determine the binary investment decisions, fixing these decisions to satisfy the first-period and exogenous NACs, and then solving the resulting problem to obtain a feasible solution (Apap and Grossmann, 2015).

4.1. Oilfield Planning Example

We consider a modified form of the MILP described in Gupta and Grossmann (2014) (see Case (i)) for maximizing the total expected NPV in the development planning of an offshore oilfield. There are 3 oilfields, 3 potential Floating Production Storage and Offloading vessels (FPSOs), and 9 possible field-FPSO connections. A total of 30 wells can be drilled over a 5-year planning horizon: 7 for field I, 11 for field II, and 12 for field III. There is also a 3-year lead time for FPSO construction and a 1-year lead time for FPSO expansion. Fields II and III have a known recoverable oil volume (size); however, the size of field I is uncertain.

Specifically, there are 2 possible realizations for the size of field I, both with equal probabilities. Note that this is an *endogenous* uncertainty, since the size cannot be realized until we drill the field and begin producing from it. The oil and gas prices are also uncertain, with 2 possible realizations with equal probabilities in each time period. These are *exogenous* uncertainties, since the oil and gas prices are market values that will be realized automatically in each time period. These prices are assumed to be correlated. The network superstructure for this problem instance is shown in Fig. 8a.

We have 2 possible combinations of realizations for the endogenous parameters and 32 possible combinations of realizations for the exogenous parameters. By considering all possible combinations of realizations of the uncertain parameters, we generate 64 scenarios with equal probabilities. This gives rise to a 6-stage, mixed-integer linear stochastic programming problem with 333,249 constraints, 70,465 continuous variables, and 7,360 binary variables. After applying the theoretical reduction properties (see Apap and Grossmann (2015)), there are 124,980 constraints (a 62% reduction), 70,465 continuous variables, and 7,000 binary variables. The problem was modeled in GAMS 24.3.3 and solved with CPLEX 12.6.0.1 on a machine with a 2.93 GHz Intel Core i7 CPU and 12 GB of RAM.



(a) Superstructure

(b) Structure for best feasible solution

Fig. 8. Network structure for the oilfield development planning problem.[2]

Table 1 summarizes the results for the example problem. In the case of solving the reduced model directly, the optimality gap cannot be improved past 50% after more than 11 hours. In contrast, the sequential scenario decomposition heuristic (SSD) finds a high-quality feasible solution ($7.166 billion) in only 41 seconds. The Lagrangean decomposition (LD) algorithm uses this lower bound and finds a high-quality upper bound ($7.180 billion) in 14 seconds. This implies that the SSD solution is within 0.20% of the optimum. Furthermore, note that we obtain this information in less than one minute of CPU time.

Table 1. Numerical results for the oilfield development planning problem.

Problem Type	Total Expected NPV ($\$10^9$)

[2] FPSO images from www.rigzone.com.

	Lower Bound	Upper Bound	Optimality Gap	Solution Time (s)
Reduced Model	6.968	10.495	50.61%	40,562
SSD	7.166	-	0.20%	41
LD	-	7.180		14

The network structure corresponding to the best feasible solution ($7.166 billion, as obtained by the SSD heuristic) is shown in Fig. 8b. This solution indicates that we begin installing all necessary infrastructure in the first year. This includes FPSO I and FPSO II, as well as the following field-FPSO connections: field I to FPSO I, field II to FPSO I, and field III to FPSO II. Notice that due to the inherent risk in the size of field I, FPSO I is shared among fields I and II rather than devoting a separate FPSO solely to field I.

The corresponding drilling schedule is shown in Fig. 9. Since it takes 3 years for the FPSOs to be fully operational, drilling cannot begin until the fourth year. For Field II, we drill 10 wells in year 4 and 1 well in year 5. Similarly for Field III, we drill 10 wells in year 4 and 2 wells in year 5. For Field I, we instead wait until year 5 and then drill 7 wells. The strategy here is to drill fields of known sizes first (as this carries less risk), and then drill the field with an uncertain size.

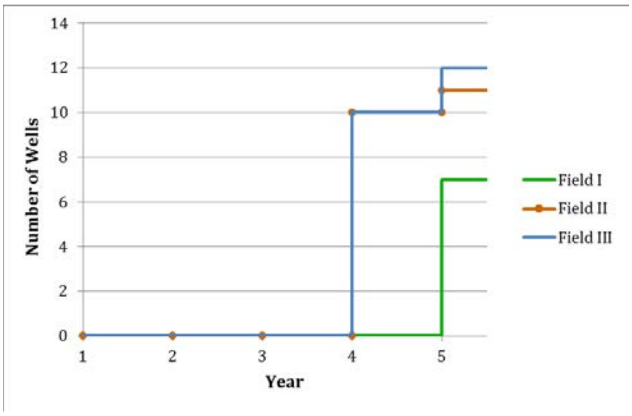


Fig. 9. Drilling schedule for the best feasible solution of the oilfield development planning problem.

5. Data-Driven Approaches to Scenario Generation

Before solving a robust, chance constrained or stochastic optimization model in practice, one has to choose a proper model of the uncertainty. This is motivated by the fact that the quality of such model directly impacts the quality of the solution of the optimization problem. In this section, we discuss methods for uncertainty modeling that use historical and forecast data for scenario-based optimization frameworks (e.g., stochastic programming).

Scenario trees are discrete representations of the probable outcomes of uncertain parameters, which are continuous random variables in most applications in process systems engineering. Also, a scenario tree is an input to a stochastic programming model. Therefore, systematically generating scenario trees that more accurately capture the true, but generally unknown structure of the uncertainty improves the quality of the solution to the original stochastic problem (see Chapter 4 in King and Wallace (2012)). Given an initial structure of the tree (number of stages and number of branches from each node), data-driven scenario tree generation methods directly use available data to specify the probabilities of the branches and values of the outcomes.

In this section, we discuss the property-matching method of Høyland and Wallace (2001) and an extension that addresses the potential under-determination of such method. The property-matching method aims at generating scenario trees by matching statistical properties (e.g., moments and co-moments) calculated from the tree to the respective properties estimated from actual data (historical or forecast). Specifically, generating a scenario tree requires solving an optimization problem in which the objective function is an error measure (deviation of properties from the tree and data), and the decision variables are branch probabilities and node values. Consider the case of a two-stage scenario tree with N scenarios that is used to model a single uncertain parameter (see Fig. 10). The scenario probabilities and node values are represented by the vectors p and x, respectively.

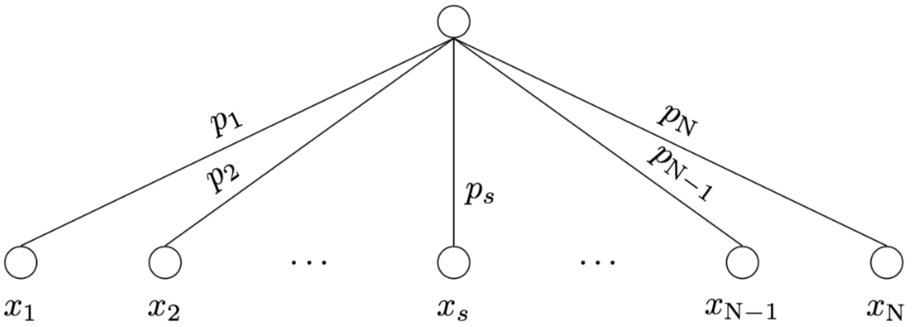


Fig. 10. Two-stage scenario tree for one uncertain parameter.

The property-matching optimization model is given in Eq. (16). The set of statistical properties to be matched is denoted by J, and the target value for property j is given by Sval_j. Also, $f_j(\cdot,\cdot)$ is the mathematical expression of statistical property j calculated from the tree, and w_j is a weight vector.

$$\min_{x,p} \sum_{j \in J} \mathrm{w}_j \cdot \left(f_j(x,p) - \mathrm{Sval}_j\right)^2 \tag{16}$$

$$\text{s.t.} \quad \sum_{s=1}^{\mathrm{N}} p_s = 1$$

$$p_s \in [0,1] \qquad \forall\, s = 1, \dots, \mathrm{N}$$

In the original Moment Matching Problem (MMP) proposed by Høyland and Wallace (see also Gülpınar, Rustem, and Settergren (2004)), the target statistical properties are moments (mean, variance, skewness, and kurtosis) and covariance. However, as discussed by Høyland and Wallace (2001), this formulation may be over- or under-determined depending on the number of branches, targets, and uncertain parameters. In fact, it is not uncommon to have an under-specified MMP, since the number of optimization variables (outcome values and their probabilities) may easily exceed the number of targets as the number of scenarios increases. We present conditions for mitigating the potential ill-posedness of property-matching formulations in equations (18) and (19). To overcome this under-specification issue and to allow more scenarios to be considered, a distribution-matching approach has been proposed and is briefly described as follows (Calfa et al., 2014).

In addition to matching (co-)moments, points of the (empirical or estimated) marginal cumulative distribution function (CDF) are also matched, thus increasing the number of target values and balancing the difference between the number of variables and data points. We call it the Distribution Matching Problem (DMP), and its L^2-norm formulation is given in equation (17). Both MMP and DMP are nonconvex nonlinear programming models. The following notation applies: the uncertain data are indexed by $i \in I$, which denotes the entity of an uncertain parameter (for example, a product; $k \in K = \{1, 2, 3, 4\}$ is the index of the first four moments; the moments calculated from the tree are denoted by variables $m_{i,k}$ and the ones calculated from the data are denoted by parameters $\mathrm{M}_{i,k}$; the covariance calculated between entity i and i' from the tree and the data are denoted by $c_{i,i'}$ and $\mathrm{C}_{i,i'}$, respectively.

$$\min_{x,p} \sum_{i\in I} \sum_{k\in K} \mathrm{w}_{i,k} \left(m_{i,k} - \mathrm{M}_{i,k}\right)^2 + \sum_{\substack{(i,i')\in I \\ i<i'}} \mathrm{w}_{i,i'} \left(c_{i,i'} - \mathrm{C}_{i,i'}\right)^2 + \sum_{i\in I} \sum_{s=1}^{\mathrm{N}} \omega_{i,s} \delta_{i,s}^2 \tag{17a}$$

$$\text{s.t.} \quad \sum_{s=1}^{\mathrm{N}} p_s = 1 \tag{17b}$$

$$m_{i,1} = \sum_{s=1}^{\mathrm{N}} x_{i,s} p_s \qquad \forall\, i \in I \tag{17c}$$

$$m_{i,k} = \sum_{s=1}^{\mathrm{N}} \left(x_{i,s} - m_{i,1}\right)^k p_s \qquad \forall\, i \in I,\ k > 1 \tag{17d}$$

$$c_{i,i'} = \sum_{s=1}^{\mathrm{N}} (x_{i,s} - m_{i,1})(x_{i',s} - m_{i',1}) p_s \quad \forall\, (i,i') \in I,\ i < i' \tag{17e}$$

$$x_{i,s} \in \left[x_{i,s}^{\mathrm{LB}}, x_{i,s}^{\mathrm{UB}}\right] \quad \forall\, i \in I,\ s = 1, \dots, \mathrm{N} \tag{17f}$$

$$p_s \in [0,1] \quad \forall\, s = 1, \dots, \mathrm{N} \tag{17g}$$

$$\widehat{CDF}(x_{i,s}) - \sum_{s'=1}^{s} p_{s'} = \delta_{i,s} \quad \forall\, i \in I,\ s = 1, \dots, \mathrm{N} \tag{17h}$$

$$x_{i,s} \leq x_{i,s+1} \quad \forall\, i \in I,\ s = 1, \dots, \mathrm{N} - 1 \tag{17i}$$

where the weighted squared error between the statistical properties calculated from the tree and inferred from the data, is minimized in (17a). Constraints (17b) ensure that the probabilities of outcomes add up to 1, (17c) represent the calculation of the first moment (mean), constraints (17d) represent the calculation of higher-order central moments, constraints (17e) are the expressions for the covariance, and $\mathrm{w}_{i,k} = \mathrm{w}'_{i,k}/\mathrm{M}^2_{i,k}$ and $\mathrm{w}_{i,i'} = \mathrm{w}'_{i,i'}/\mathrm{C}^2_{i,i'}$, where $\mathrm{w}'_{i,k}$ and $\mathrm{w}'_{i,i'}$ are weights, which can be chosen arbitrarily. The bounds on the decision variables x and p are represented in constraints (17f) and (17g), respectively. The variables $\delta_{i,s}$ represent the deviations with respect to the empirical or estimated CDF, which is represented by the term $\widehat{CDF}(\cdot)$ in (17h). Constraints (17i) ensure that the node values are ordered due to the cumulative information being matched.

The original MMP only includes constraints (17b) – (17g), and its objective function only contains the first two terms in equation (17a) regarding the moments and covariance. The under-specification of both MMP and DMP formulations is analyzed as follows. For the MMP, the number of variables is $|I| \cdot \mathrm{N} + \mathrm{N} - 1 = \mathrm{N}(|I| + 1) - 1$. The number of data points ($|K|$ moments and covariances, or conditions to be matched) is $|I| \cdot |K| + \frac{|I|(|I|-1)}{2}$. The MMP is well-posed if the number of data points is at least the number of variables, i.e.,

$$|I| \cdot |K| + \frac{|I|(|I| - 1)}{2} \geq \mathrm{N}(|I| + 1) - 1 \tag{18}$$

We perform the same analysis for the DMP. The number of variables is the same as in the MMP, but the number of data points (conditions to be matched) is increased by $|I| \cdot \mathrm{N}$. Thus, the condition for well-posedness of the DMP is given by,

$$|I| \cdot |K| + \frac{|I|(|I| - 1)}{2} \geq \mathrm{N} - 1 \tag{19}$$

Conditions (18) for the MMP and (19) for the DMP can be illustrated with the following example. For one uncertain parameter ($|I| = 1$) and four moments ($|K| = 4$), we have that $\mathrm{N} \leq 2.5$ for the MMP and that $\mathrm{N} \leq 5$ for the DMP. Consequently, more scenarios can be used in the DMP formulation without leading to under-specified problems.

The DMP formulation in equation (17) can be used for multi-stage scenario tree generation. One approach is to solve multiple DMPs *sequentially*, from the root node to every node of the penultimate stage. The solution approach consists of two steps: **forecasting step**, where future targets are predicted and used by the **optimization step**, where the outcomes and their probabilities are obtained by solving the DMP (see details in Calfa et al., 2014).

5.1. Two-Stage Scenario Tree Generation Example

A network of three chemical plants is described in Calfa et al. (2014) (see Fig. 11). The yield of plant P1 is the uncertain parameter and the distribution of its historical data is given in Fig. 12.

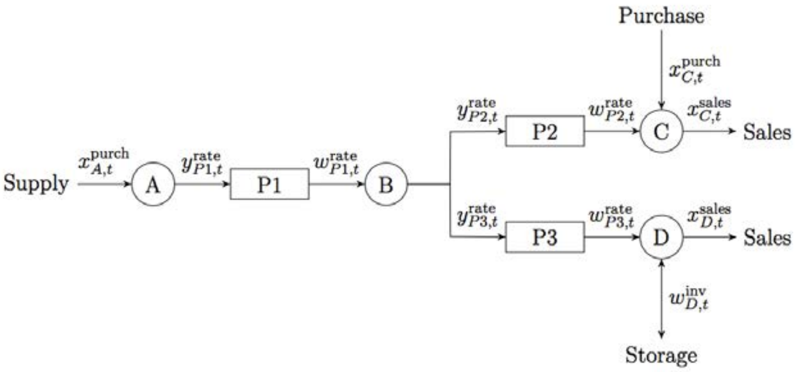


Fig. 11. Process network with uncertain yield for Plant 1.

The empirical CDF was obtained and approximated by a smooth function (generalized logistic function), i.e., $\widehat{CDF}(\cdot)$ in Eq. (17). In this example, $\mathrm{N} = 5$ scenarios were considered, and the targets include the first four moments and CDF information. The scenario generation problem is a nonlinear program with 10 variables and 14 constraints. It was modeled in AIMMS 3.13 and solved with IPOPT 3.10.1 using AIMMS' multi-start module in less than one second. Fig. 13 shows the profile of optimized scenario probabilities as well as node values of the generated scenario tree. Note that the profile captures the shape of the distribution of the original data, including tail effects.

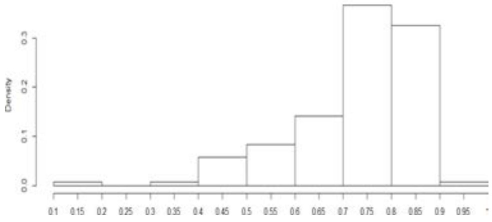


Fig. 12. Distribution of the historical data for the production yield of plant P

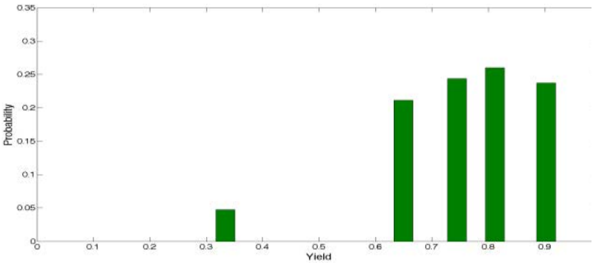


Fig. 13. Profile of scenario probabilities obtained with the

distribution-matching approach.

6. Conclusions

In this paper, we have addressed several major challenges in optimization under uncertainty for MILP models. We have shown that recourse can be accounted for in robust optimization with linear decision rules, which has the effect of producing less conservative solutions as was shown in the example of an air separation plant with interruptible load. To effectively solve two-stage stochastic optimization problems, we have shown the importance of developing tailored solution strategies; e.g., by reducing the number of scenarios, tightening the MILP formulation, and applying decomposition schemes like the L-shaped method with multiple cuts. This was illustrated in the case of supply chains under disruptions. We have also shown that both exogenous and endogenous uncertainties can be effectively handled in multi-stage stochastic programming by relying on theoretical properties to reduce the number of non-anticipativity constraints, and applying Lagrangean decomposition coupled with fast heuristics. This was illustrated in an oilfield planning problem. Finally, we have shown that to avoid assigning arbitrary probabilities and outcomes in scenario trees, historical data can be incorporated to generate these scenarios by using moment matching supplemented by matching cumulative distribution function values. This was illustrated with a production planning problem.

All of the examples in this paper have demonstrated the progress that has been made in this area, although the models have been restricted to MILP models. It is clear that major challenges remain in this area that we have not addressed in this paper. One of these challenges is the effective solution of stochastic MINLP problems (Li et al., 2011; Tarhan et al., 2013), in which the computational requirements are much higher than in MILP models. Another one is the effective incorporation of risk measures in the objective function (Applequist et al., 2000; Barbaro and Bagajewicz, 2004; Rockafellar, 2007; You et al., 2009). Finally, interpretation of results by non-expert users from optimization models under uncertainty is a major challenge given that that there are different strategies for hedging against uncertainty. Here the use of simulation would seem to be one promising avenue (You et al., 2009).

Acknowledgments. The authors would like to acknowledge financial support from NSF Grant No. 1159443, Praxair, The Dow Chemical Company, the ExxonMobil Upstream Research Company, and the Center for Advanced Process Decision-making at Carnegie Mellon University.

References

Ahmed, S., & Garcia, R. (2003). Dynamic Capacity Acquisition and Assignment under Uncertainty. Annals of Operations Research, 124(1), 267-283.

Amaran, S., Zhang, T., Sahinidis, N. V., Sharda, B., & Bury, S. J. (2015). Medium-term Maintenance Turnaround Planning under Uncertainty for Integrated Chemical Sites. To appear in Computers & Chemical Engineering.

Apap, R. M., & Grossmann, I. E. (2015). Models and Computational Strategies for Multistage Stochastic Programming under Endogenous and Exogenous Uncertainties. Submitted for publication.

Applequist, G. E., Pekny, J. F., & Reklaitis, G. V. (2000). Risk and Uncertainty in Managing Chemical Manufacturing Supply Chains. Computers & Chemical Engineering, 24(9-10), 2211-2222.

Baptista, S., Gomes, M. I., & Barbosa-Póvoa, A. P. (2012). A Two-Stage Stochastic Model for the Design and Planning of a Multi-Product Closed Loop Supply Chains. Computer Aided Chemical Engineering, 30, 412-416.

Barbaro, A., & Bagajewicz, M. J. (2004). Managing Financial Risk in Planning under Uncertainty. AIChE Journal, 50(5), 963-989.

Ben-Tal, A., Chung, B. D., Mandala, S. R., & Yao, T. (2011). Robust Optimization for Emergency Logistics Planning: Risk Mitigation in Humanitarian Relief Supply Chains. Transportation Research Part B: Methodological, 45(8), 1177-1189.

Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). Robust Optimization. Princeton University Press. Princeton, NJ.

Ben-Tal, A., Goryashko, A., Guslitzer, E., & Nemirovski, A. (2004). Adjustable Robust Solutions of Uncertain Linear Programs. Mathematical Programming, 99(2), 351-376.

Bertsimas, D., & Sim, M. (2004). The Price of Robustness. Operations Research, 52(1), 35-53.

Birge, J. R., & Louveaux, F. V. (1988). A Multicut Algorithm for Two-Stage Stochastic Linear Programs. European Journal of Operational Research, 34(3), 384-392.

Birge, J. R., 1997. Stochastic Programming Computation and Applications. INFORMS Journal on Computing, 9(2), 111-133.

Birge, J., & Louveaux, F. V. Introduction to Stochastic Programming (2nd Ed.), Ch. 3: Basic Properties and Theory. Springer: New York (NY), 2011.

Boland, N., Dumitrescu, I., & Froyland, G. (2008). A Multistage Stochastic Programming Approach to Open Pit Mine Production Scheduling with Uncertain Geology. Optimization Online, http://www.optimization-online.org/DB_FILE/2008/10/2123.pdf. Accessed September 24, 2015.

Calfa, B. A., Agarwal, A., Grossmann, I. E., & Wassick, J. M. (2014). Data-Driven Multi-Stage Scenario Tree Generation via Statistical Property and Distribution Matching. Computers & Chemical Engineering, 68(1), 7-23.

Carøe, C. C., Schultz, R. (1999). Dual Decomposition in Stochastic Integer Programming. Operations Research Letters, 24(1-2), 37-45.

Chen, X., Zhang, Y. (2009). Uncertain Linear Programs: Extended Affinely Adjustable Robust Counterparts. Operations Research, 57(6), 1469-1482.

Cheng, L., Subrahmanian, E., & Westerberg, A. W. (2003). Design and Planning under Uncertainty: Issues on Problem Formulation and Solution. Computers & Chemical Engineering, 27(6), 781-801.

Christian, B. & Cremaschi, S. (2014). Heuristic solution approaches to the pharmaceutical R&D pipeline management problem. Computers & Chemical Engineering, 74, 34-47.

Colvin, M., & Maravelias, C. T. (2008). A Stochastic Programming Approach for Clinical Trial Planning in New Drug Development. Computers & Chemical Engineering, 32(11), 2626-2642.

Colvin, M., & Maravelias, C. T. (2010). Modeling Methods and a Branch and Cut Algorithm for Pharmaceutical Clinical Trial Planning Using Stochastic Programming. European Journal of Operational Research, 203(1), 205-215.

García-Herreros, P., Wassick, J. M., & Grossmann, I. E. (2014). Design of Resilient Supply Chains with Risk of Facility Disruptions. Industrial and Engineering Chemistry Research, 53(44), 17240-17251.

Goel, V., & Grossmann, I. E. (2004). A Stochastic Programming Approach to Planning of Offshore Gas Field Developments under Uncertainty in Reserves. Computers & Chemical Engineering, 28(8), 1409-1429.

Goel, V., & Grossmann, I. E. (2006). A Class of Stochastic Programs with Decision Dependent Uncertainty. Mathematical Programming: Series B, 108(2-3), 355-394.

Gounaris, C. E., Wiesemann, W., & Floudas, C. A. (2013). The Robust Capacitated Vehicle Routing Problem Under Demand Uncertainty. Operations Research, 61(3), 677-693.

Grossmann, I. E., & Floudas, C.A. (1987). Active Constraint Strategy for Flexibility Analysis in Chemical Processes. Computers & Chemical Engineering, 11(6), 675-693.

Grossmann, I. E., Calfa, B. A., & García-Herreros, P. (2014). Evolution of Concepts and Models for Quantifying Resiliency and Flexibility of Chemical Processes. Computers & Chemical Engineering 70, 22-34.

Guignard, M., & Kim, S. (1987). Lagrangean Decomposition: A Model Yielding Stronger Lagrangean Bounds. Mathematical Programming, 39(2), 215-228.

Gupta, V., & Grossmann, I. E. (2011). Solution Strategies for Multistage Stochastic Programming with Endogenous Uncertainties. Computers & Chemical Engineering, 35, 2235-2247.

Gupta, V., & Grossmann, I. E. (2014). A New Decomposition Algorithm for Multistage Stochastic Programs with Endogenous Uncertainties. Computers & Chemical Engineering, 62, 62-79.

Gülpınar, N., Rustem, B., & Settergren, R. (2004). Simulation and Optimization Approaches to Scenario Tree Generation. Journal of Economic Dynamics and Control, 28(7), 1291-1315.

Halemane, K.P., & Grossmann, I. E. (1983). Optimal Process Design under Uncertainty. AIChE Journal, 29(3), 425-433.

Høyland, K., & Wallace, S. W. (2001). Generating Scenario Trees for Multistage Decision Problems. Management Science, 46(2), 295-307.

Ierapetritou, M. G., & Pistikopoulos, E. N. (1994). Novel Optimization Approach of Stochastic Planning Models. Industrial and Engineering Chemistry Research, 33(8), 1930-1942.

Ierapetritou, M. G., & Pistikopoulos, E. N. (1995). Design of Multiproduct Batch Plants with Uncertain Demands. Computers & Chemical Engineering, 19(S1), 627-632.

Iyer, R. R., & Grossmann, I. E. (1998). A Bilevel Decomposition Algorithm for Long-Range Planning of Process Networks. Industrial and Engineering Chemistry Research, 37(2), 474-481.

Janak, S. L., Lin, X., & Floudas, C. A. (2007). A New Robust Optimization Approach for Scheduling under Uncertainty: II. Uncertainty with Known Probability Distribution. Computers & Chemical Engineering, 31(3), 171-195.

Jonsbråten, T. W. (1998). Optimization Models for Petroleum Field Exploitation, Ph.D. Thesis, Norwegian School of Economics and Business Administration, Norway.

Jonsbråten, T. W., Wets, R. J. B., & Woodruff, D. L. (1998). A Class of Stochastic Programs with Decision Dependent Random Elements. Annals of Operations Research, 82, 83-106.

Jung, J. Y., Blau, G., Pekny, J. F., Reklaitis G. V., & Eversdyk, D. (2004). A Simulation Based Optimization Approach to Supply Chain Management under Demand Uncertainty. Computers & Chemical Engineering, 28(10), 2087-2106.

King, A. J., & Wallace, S. W. (2012). Modeling with Stochastic Programming. Springer Series in Operations Research and Financial Engineering. New York, NY. Springer Science+Business Media New York.

Kuhn, D., Wiesemann, W., & Georghiou, A. (2011). Primal and Dual Linear Decision Rules in Stochastic and Robust Optimization. Math Programming, 130(1), 177-209.

Li, P., Arellano-Garcia, H., & Wozny, G. (2008). Chance Constrained Programming Approach to Process Optimization under Uncertainty. Computers & Chemical Engineering, 32(1-2), 25-45

Li, X., Tomasgard, A., & Barton, P. I. (2011). Nonconvex Generalized Benders Decomposition for Stochastic Separable Mixed-Integer Nonlinear Programs. Journal of Optimization Theory and Applications, 151(3), 425-454.

Li, Z., & Ierapetritou, M. (2008a). Process Scheduling under Uncertainty: Review and Challenges. Computers & Chemical Engineering, 32(4-5), 715-727

Li, Z., & Ierapetritou, M. (2012). Capacity Expansion Planning through Augmented Lagrangian Optimization and Scenario Decomposition. AIChE Journal, 58(3), 871-883.

Li, Z., & Ierapetritou, M., (2008b). Robust Optimization for Process Scheduling Under Uncertainty. Industrial and Engineering Chemistry Research, 47(12), 4148-4157.

Li, Z., Ding, R., & Floudas, C. A. (2011). A Comparative Theoretical and Computational Study on Robust Counterpart Optimization: I. Robust Linear Optimization and Robust Mixed Integer Linear Optimization. Industrial and Engineering Chemistry Research, 50(18), 10567-10603.

Li, Z., Tang, Q., & Floudas, C. A. (2012). A Comparative Theoretical and Computational Study on Robust Counterpart Optimization: II. Probabilistic Guarantees on Constrain Satisfaction. Industrial and Engineering Chemistry Research, 51(19), 6769-6788.

Lin, X., Janak, S. L., & Floudas, C. A. (2004). A New Robust Optimization Approach for Scheduling under Uncertainty: I. Bounded Uncertainty. Computers & Chemical Engineering, 28(6-7), 1069-1085

Magnanti, T. L., & Wong, R. T. (1981). Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. Operations Research, 29(3), 464-484.

Mitra, S., Grossmann, I. E., Pinto, J. M., & Arora, N. (2012). Robust Scheduling under Time-Sensitive Electricity Prices for Continuous Power-Intensive Processes. Proceedings of the Foundations of Computer-Aided Process Operations 2012.

Pintarič, Z. N., & Kravanja, Z. (2004). A Strategy for MINLP Synthesis of Flexible and Operable Processes. Computers & Chemical Engineering, 28(6-7), 1105-1119.

Rockafellar, R. T. (2007). Coherent Approaches to Risk in Optimization Under Uncertainty. INFORMS Tutorials in Operations Research (March 2015), 38-61.

Ruszczyński, A. (1997). Decomposition Methods in Stochastic Programming. Mathematical Programming, 79(1-3), 333-353.

Sahinidis, N. V. (2004). Optimization under Uncertainty: State-of-the-Art and Opportunities. Computers & Chemical Engineering, 28(6), 971-983.

Sand, G., & Engell, S. (2004). Modeling and Solving Real-Time Scheduling Problems by Stochastic Integer Programming. Computers & Chemical Engineering, 28(6-7), 1087-1103.

Schultz, R. (2003). Stochastic Programming with Integer Variables. Mathematical Programming, 97(1), 285-309.

Subrahmanyam, S., Pekny, J. F., & Reklaitis, G. V. (1994). Design of Batch Chemical-Plants under Market Uncertainty, Industrial and Engineering Chemistry Research, 33(11), 2688-2701.

Swaney, R. E., & Grossmann, I. E. (1985). An Index for Operational Flexibility in Chemical Process Design. AIChE Journal, 31(4), 621-630.

Tarhan, B. & Grossmann, I. E. (2008). A Multistage Stochastic Programming Approach with Strategies for Uncertainty Reduction in the Synthesis of Process Networks with Uncertain Yields. Computers and Chemical Engineering, 32, 766-788.

Tarhan, B., Grossmann, I. E., & Goel, V. (2009). Stochastic Programming Approach for the Planning of Offshore Oil or Gas Field Infrastructure under Decision-Dependent Uncertainty. Ind. Eng. Chem. Res., 48(6), 3078-3097.

Tarhan, B., Grossmann, I. E., & Goel, V. (2013). Computational Strategies for Non-convex Multistage MINLP Models with Decision-Dependent Uncertainty and Gradual Uncertainty Resolution. Annals of Operations Research, 203(1), 141-166.

Tsiakis, P., Shah, N., & Pantelides, C. C. (2001). Design of Multi-Echelon Supply Chain Networks under Demand Uncertainty. Industrial and Engineering Chemistry Research, 40(16), 3585-3604.

Van Slyke, R. M., & Wets, R. (1969). L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Programming. SIAM Journal on Applied Mathematics, 17(4), 638-663.

Verderame, P. M., & Floudas, C. A. (2009). Operational Planning of Large-Scale Industrial Batch Plants under Demand Due Date and Amount Uncertainty: I. Robust Optimization Framework. Industrial and Engineering Chemistry Research, 48(15), 7214-7231.

Vujanic, R., Mariéthoz, S., Goulart, P., & Morari, M. (2012). Robust Integer Optimization and Scheduling Problems for Large Electricity Consumers. Proceedings of the 2012 American Control Conference, 3108-3113.

You, F., Wassick, J. M., & Grossmann, I. E. (2009). Risk Management for a Global Supply Chain Planning under Uncertainty: Models and Algorithms, AIChE Journal, 55(4), 931-946.

Zhang, Q., Heuberger, C. F., Grossmann, I. E., Sundaramoorthy, A., & Pinto, J. M. (2015a). Air Separation with Cryogenic Energy Storage: Optimal Scheduling Considering Electric Energy and Reserve Market Participation. AIChE Journal, 61(5), 1547-1558.

Zhang, Q., Lima, R. M., & Grossmann, I. E. (2015). On the Relation Between Flexibility Analysis and Robust Optimization for Linear Systems. Submitted for publication.

Zhang, Q., Morari, M. F., Grossmann, I. E., Sundaramoorthy, A., & Pinto, J. M. (2015b). An Adjustable Robust Optimization Approach to Scheduling of Continuous Industrial Processes Providing Interruptible Load. Computers & Chemical Engineering, 86, 106-119.