Computational Strategies for Non-convex Multistage MINLP Models with Decision-Dependent Uncertainty and Gradual Uncertainty Resolution

Bora Tarhan,* Ignacio E. Grossmann,†
Department of Chemical Engineering, Carnegie Mellon University
Pittsburgh, PA 15213

Vikas Goel‡
ExxonMobil Upstream Research Company, Houston, TX 77098

Abstract

In many planning problems under uncertainty the uncertainties are decision-dependent and resolve gradually depending on the decisions made. In this paper, we address a generic non-convex MINLP model for such planning problems where the uncertain parameters are assumed to follow discrete distributions and the decisions are made on a discrete time horizon. In order to account for the decision-dependent uncertainties and gradual uncertainty resolution, we propose a multistage stochastic programming model in which the non-anticipativity constraints in the model are not prespecified but change as a function of the decisions made. Furthermore, planning problems consist of several scenario subproblems where each subproblem is modeled as a nonconvex mixed-integer nonlinear program. We propose an efficient solution strategy that combines global optimization and outer-approximation in order to optimize the planning decisions. We apply this generic problem structure and the proposed solution algorithm to several planning problems to illustrate the efficiency of the proposed method.

Keywords: decision making under uncertainty, decision dependent uncertainty, gradual uncertainty resolution, multistage stochastic programming, non-convex mixed integer nonlinear program, global optimization, outer-approximation, oil or gas field exploration, synthesis of process networks

1 Introduction

Stochastic programming is a mathematical programming framework for modeling optimization problems that involve uncertainty in the data which are represented by probability distributions. There are two broad

*E-mail: btarhan@andrew.cmu.edu
†To whom all correspondence should be addressed. E-mail: grossmann@cmu.edu
‡E-mail: vikas.goel@exxonmobil.com

classes of stochastic programming problems: chance constrained There are two broad classes of stochastic programming problems: chance constrained (Charnes and Cooper, 1963) and problems with recourse (Birge and Louveaux, 1997). In this work we focus on ”multistage stochastic programming with recourse” and for the sake of simplicity, we will refer to it throughout the paper shortly as ”stochastic program”. The fundamental idea behind stochastic programming is the concept of recourse which is the ability of the decision-maker to take corrective action after a random event has taken place over a sequence of stages.

In this paper the uncertain parameters are assumed to follow discrete probability distributions and the planning horizon consists of a fixed number of decision points. Using these two assumptions, the stochastic process can be represented using scenario trees. Figure 1 is a standard representation of a scenario tree having one uncertain parameter (ξ_t) with two discrete values in two time periods, which leads to four scenarios. Uncertain parameters ξ_1 and ξ_2 reveal at the end of first and second time periods respectively.

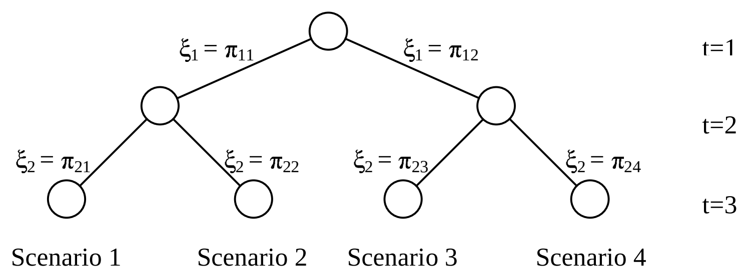

Figure 1: Scenario tree with uncertain parameters ξ_1 and ξ_2.

In a standard scenario tree, each node for time period t represents a possible state of the system at that time period. Each arc represents the possible transition from one state in time period t to another state in time period $t+1$. A path from the root node to a leaf node represents a scenario. Thus, a scenario is a combination of possible uncertain parameters in each of the time periods which in turn define a stage.

Figure 2 is an alternative representation of the scenario tree in Figure 1, proposed by Ruszczyński (1997). In this representation each scenario is denoted by a set of unique nodes. The horizontal lines connecting nodes in time period t indicate that these nodes have the same information in that time period, and therefore are indistinguishable. These non-anticipativity constraints suggest that decisions cannot be based on knowledge that will be revealed in the future. The horizontal lines reduce the tree in Figure 2 to the one shown in Figure 1. For modeling the problem, scenario trees presented in Figure 2 will be considered in order to explicitly handle the non-anticipativity constraints and incorporate decision dependent uncertainties (Tarhan and Grossmann, 2008).

(SP) is a standard linear stochastic program with $|T|$ time periods and $|S|$ scenarios, modeled using the alternative scenario tree proposed by Ruszczyński (1997). The parameters p^s represent the probability of scenario s, while the variables x_t^s represent variables for time period t in scenario s. Eq. (1.1) corresponds to the objective of maximizing the expectation of an economic criterion (e.g. net present value). Eq. (1.2)

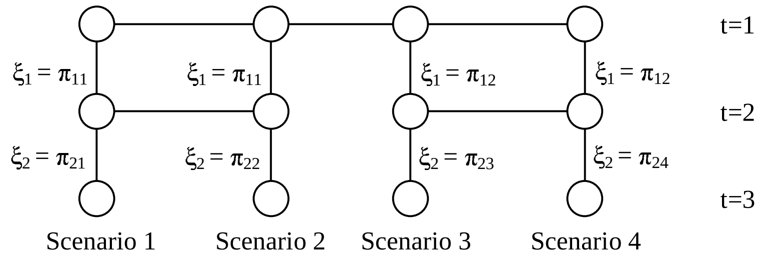

Figure 2: Alternative scenario tree with uncertain parameters ξ_1 and ξ_2.

represents the multi-period constraints for each scenario s. Eq. (1.3) represents the non-anticipativity constraints which state that decisions x_t^s and $x_t^{s'}$ must be identical if scenario pair (s, s') is indistinguishable in time period t. These non-anticipativity constraints correspond to the horizontal lines connecting different nodes in Figure 2. Eq. (1.4) represents the bounds and integrality constraints on the variables.

$$(SP) \qquad \max \sum_{s \in S} p^s \sum_{t \in T} c_t x_t^s \tag{1.1}$$

$$\sum_{\tau \leq t} A_\tau^s x_\tau^s \leq a_t^s \qquad \forall (t, s) \tag{1.2}$$

$$x_t^s = x_t^{s'} \qquad \forall \left(t, s, s'\right) \in N \tag{1.3}$$

$$x_t^s \in \chi_t^s \qquad \forall (t, s) \tag{1.4}$$

One important aspect of this paper is decision-dependent uncertainty. According to Jonsbraten (1998), uncertainty in stochastic programming problems can be divided into two classes: exogenous uncertainty and endogenous uncertainty. Most previous work in the literature deals with problems with exogenous uncertainty (e.g. demands) where the optimization decisions cannot influence the stochastic process. Reviews of previous work on problems with exogenous uncertainty can be found in (Sahinidis, 2004; Schultz, 2003). Problems where stochastic processes are affected by decisions are said to possess endogenous uncertainty. According to Goel and Grossmann (2006), decisions affect the stochastic process in at least two ways: decisions can alter the probability distributions (type 1), or decisions can act to discover more accurate information (type 2). In this paper, we focus on the latter type of endogenous uncertainty where the decisions act to resolve uncertainty. Literature on the class of problems that deal with endogenous uncertainty is limited. The only papers that we are aware of are Pflug (1990); Jonsbraten et al. (1998); Ahmed (2000); Viswanath et al. (2004); Held and Woodruff (2005); Goel and Grossmann (2004, 2006); Goel (2005); Goel et al. (2006); Tarhan and Grossmann (2008); Boland et al. (2008); Solak (2007). Detailed discussion on these papers can be found in Tarhan and Grossmann (2008) and Tarhan et al. (2009).

Boland et al. (2008) applies multistage stochastic programming to open pit mine production scheduling, which is modeled as a mixed-integer linear program. They consider endogenous uncertainty (type 2) where the excavation decisions resolve uncertainty in geology immediately. They follow a similar approach as Goel and Grossmann (2006) for modeling the problem, with the exception of eliminating some of the binary variables used in the general formulation to represent conditional non-anticipativity constraints. Furthermore, they solve the model in full-space without using any decomposition algorithm.

It is more convenient to represent the endogenous type of uncertainty using the alternative scenario tree in Figure 2 where the non-anticipativity constraints are explicitly handled as functions of decisions. For instance, if the decisions at time period 1 and 2 are such that all scenarios are indistinguishable at time period 2 and some are distinguishable at time period 3 then the corresponding scenario tree will be as shown in Figure 3. This is different from the tree in Figure 2 where scenarios 2 and 3 at time period 2, and all scenarios at time period 3 are distinguishable. Therefore, in case of endogenous uncertainty, the set of non-anticipativity constraints (N) in eq. (1.3) is not fixed but it is a function of the decisions (x_t^s).

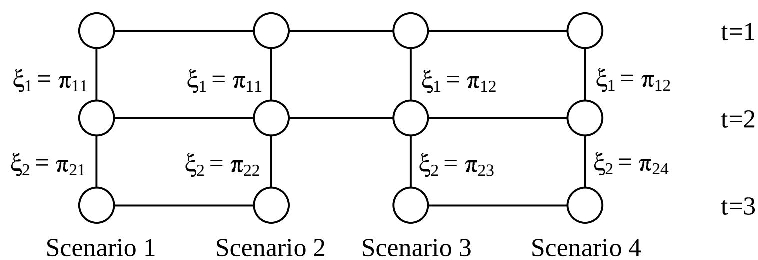

Figure 3: Scenario tree with different uncertainty resolution scheme.

Another aspect that will be addressed in this paper is gradual resolution of endogenous uncertainty over time. Recent publications in the area, Goel and Grossmann (2006) and Boland et al. (2008), consider that the endogenous uncertainty resolves immediately when the decisions to resolve the uncertainties are made. Different from the previous work, we consider the case where such uncertainty resolves gradually depending on the decisions made. For instance the uncertainty in the yield of a process can resolve as more production is performed. Since the resolution of yield is dependent on the operating decisions, it is of endogenous type. It is assumed that uncertainty in the yield resolves gradually depending on the operating decisions made in each time period. This brings some changes to the model, underlying scenario tree, and the non-anticipativity constraints. The literature about this subject in the context of planning problems is also very limited. The only literature that we know is Stensland and Tjøstheim (1991); Jonsbraten (1998); Dias (2002); Tarhan and Grossmann (2008). Detailed discussion on these papers can also be found in Tarhan and Grossmann (2008). Recently, Solak (2007) considers the stochastic programming models in which times of uncertainty realizations are dependent on the decisions made and uncertainty resolves gradually. They use a similar modeling approach presented in Tarhan and Grossmann (2008). Specifically, they consider the project portfolio optimization problem that deals with the selection of research and development projects and determination of optimal resource allocations. They use sample average approximation method (Kleywegt et al., 2002) for solving the problem, where the sample problems are solved through Lagrangian relaxation and heuristics.

Harrison (2007) uses a different approach for optimizing two-stage decision making problems under uncertainty. Some of the uncertainty is assumed to resolve after the observation of the outcome of the first stage decision. The author develops a new method called Bayesian Programming, where the necessary integrals are approximated using Markov Chain Monte Carlo simulations, and simulated annealing type of meta-heuristic are used to optimize the decisions.

The outline of this paper is as follows. In Section 2 we present the generic mathematical model for the class of problems under consideration. In Section 3 the proposed solution approach is explained. Section 4 compares the results found by the proposed solution approach, and discusses the results on synthesis of process networks and planning of offshore oil or gas field infrastructure problems.

2 Mathematical model

In this chapter, only generic variable names are used in the explanation of the model for the problems under consideration. The generic variable names are selected based on whether a variable is decision, state, or recourse variable. The decision, state, and recourse variables are denoted by vectors d, x and u with dimension n_d, n_x, and n_u respectively. The first $n'_d \leq n_d$, $n'_x \leq n_x$, $n'_u \leq n_u$ elements of these vectors are restricted to be discrete variables. Also, w and z are logic/binary variables used for defining the conditional non-anticipativity constraints.

The planning horizon is divided into $|T|$ planning periods. The sequence of decisions is as follows: the decision variables d_t are implemented in the beginning of time period are t which is followed by the resolution of uncertainty. The state variables x_t are automatically calculated when the decisions are fixed. At the end of time period t, some recourse decisions u_t are implemented. (See Figure 4)

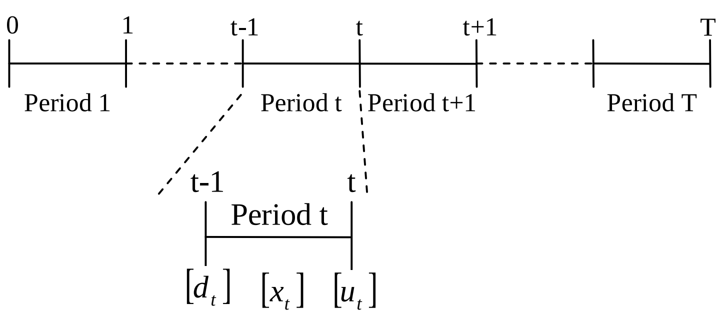

Figure 4: Representation of planning horizon.

Based on the above definitions, the proposed multistage stochastic MINLP model is as follows:

$$(P) \qquad max \quad \sum_{t\in T}\sum_{s\in S} p^s \, f\left(d_t^s, u_t^s\right) \tag{2.1}$$

$$h\left(d_t^s, x_t^s, u_t^s\right) \le 0 \qquad \forall t \in T, \forall s \in S \tag{2.2}$$

$$g\left(d_t^s, x_t^s, u_t^s\right) \le 0 \qquad \forall t \in T, \forall s \in S \tag{2.3}$$

$$w_t^{l,s} \Leftrightarrow \Phi\left(d_t^s, x_t^s, u_t^s\right) \qquad \forall t \in T, \forall s \in S, \forall l \in L \tag{2.4}$$

$$z_t^{s,s'} \Leftrightarrow \Psi\left(w_t^s\right) \qquad \forall t \in T, \forall \left(s, s'\right) \in M_q, \forall q \in Q \tag{2.5}$$

$$\begin{bmatrix} z_t^{s,s'} \\ d_{t+1}^s = d_{t+1}^{s'} \end{bmatrix} \vee \left[\neg z_t^{s,s'}\right] \qquad \forall \left(s, s', t\right) \in N_C, \forall \left(s, s'\right) \in M_q, \forall q \in Q \tag{2.6}$$

$$d_t^s = d_t^{s'} \qquad \forall \left(s, s', t\right) \in N_I \tag{2.7}$$

$$d_t^s, x_t^s, u_t^s \in \chi_t^s \qquad \forall t \in T, \forall s \in S \tag{2.8}$$

$$z_t^{s,s'}, w_t^s \in \{True, False\} \qquad \forall \left(s, s', t\right) \in N_C \tag{2.9}$$

In eq. (2.1), the objective is to maximize the expected net present value, which is a linear or non-linear function of decision and state variables. Eq. (2.2) is a generic multi period linear constraint that relates the decision, state, and recourse variables for every scenario. Similarly, eq. (2.3) is a generic non-linear constraint for every scenario. Eq. (2.4) relates the binary/logic variables w with the discrete and continuous decision, and state variables. These w variables are used in eq. (2.5) to model problem specific uncertainty resolution rules to determine if scenario pair (s, s') is indistinguishable or not. There are two types of non-anticipativity constraints: initial (N_I) and conditional (N_C). Eq. (2.7) is the initial non-anticipativity constraints that hold regardless of any decision taken. Note that the initial non-anticipativity constraints include not only the first period non-anticipativity constraints but also some other subsequent period non-anticipativity constraints that must hold because of gradual uncertainty resolution. The conditional non-anticipativity constraints in the disjunction (eq. (2.6)) are included into the model if two scenarios are indistinguishable at the end of time period t ($z_t^{s,s'}$ is true). The conditional non-anticipativity constraints dictate that the decisions at the beginning of time period $t+1$ in scenarios s and s' must be identical. Otherwise they are not imposed on the feasible space. As proved in Goel and Grossmann (2006), it is enough to consider a subset of scenario pairs that differ in only one uncertain parameter. Therefore, in eqs. (2.5) and (2.6) each such scenario pair (s, s') will be an element of exactly one of the sets M_q, $\forall q \in Q$. Note that in eqs. (2.4) and (2.5), vector w_t^s contains $w_t^{l,s}$ for each resolution level $l \in L$ (e.g. in Figure 6 $L = \{1, 2, 3\}$). Eqs. (2.8)–(2.9) represent the variables properties and integrality requirements.

Note that conditional non-anticipativities are not imposed on recourse variables u_t. This is different than Goel and Grossmann (2006) where the uncertainty was assumed to resolve instantaneously. In case of gradual uncertainty, although the decision maker observes some output from the system, the decision maker may not differentiate scenarios. For instance, let us assume there is a process which takes a raw material and generates a final product (Figure 5). The demand for final product has to be satisfied exactly and in case of shortage, the final product can be bought from the market. The decision at the beginning of each time period is the amount of raw material to buy and the recourse at the end of each period is the amount of final product to buy from the market to satisfy the shortage.

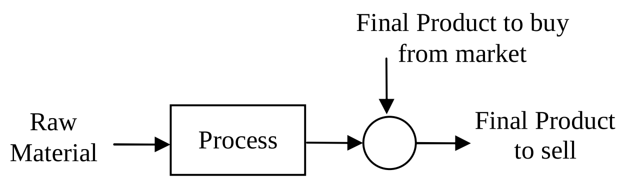

Figure 5: Simple process converting raw material to final product.

Assume there is uncertainty in the yield of the process and it requires two years of production to resolve the uncertainty, which has four possible values, values, θ^1, θ^2, θ^3, θ^4. After one year of production the decision maker may observe that the yield is on the low side (θ^1, θ^2) or on the high side (θ^3, θ^4). At this partial resolution stage, the decision maker has to give identical decisions for the amount of raw material to purchase for the second year for each indistinguishable scenario groups (i.e. scenarios (1-4) and (5-8)) as shown in the scenario tree in Figure 6. The decision maker cannot distinguish the scenarios within the scenario groups (1-4) and (5-8) at partial resolution stage since the decision maker needs one more year to know the exact yield. On the other hand, the recourse decision (amount of final product to buy from the market) at the end of first year has to be different among some of the indistinguishable scenarios (e.g. (1-2) and (3-4)) because the amount of production varies due to different yields whereas the demand for the final product stays the same. Therefore, when two scenarios are indistinguishable we impose only the decision variables, and not the recourse variables, in the disjunction in eq. (2.6).

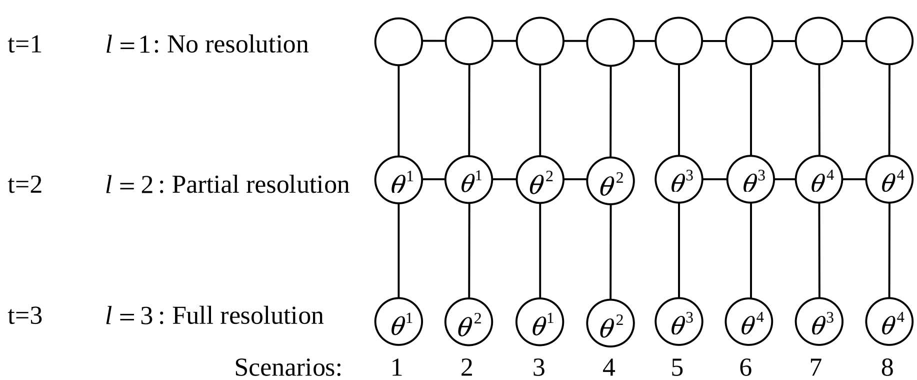

Figure 6: Scenario tree representation of gradual uncertainty resolution in yield.

To illustrate the constraints (2.2)-(2.5), we give specific examples from the synthesis of process networks problem (details in Appendix A). In the generic model (P), eq. (2.2) corresponds to the linear constraints such as the material balance constraints (A.3), (A.7), (A.8), (A.9), (A.10) on every node and process. Eq. (2.3) corresponds to the non-linear constraints that are used for calculating net present value for every scenario (A.2). Eq. (2.4) corresponds to constraints (A.4) and (A.5) which state that logic variable $w_{i,t}^{1,s}$ will be true if and only if process i has not been expanded, or not run as a pilot plant until time period $t-\tau'$. Similarly, $w_{i,t}^{2,s}$ will be true if and only if process i has operated only one time period or run as a pilot plant until time period $t-\tau''$. Eq. (2.5) corresponds to constraint (A.25) which state that a pair of scenarios that is an element of $M_q, \forall q \in Q$ (see Appendix A for specific subsets used for the example problem) will be indistinguishable if and only if for each process that distinguishes the scenario pair, $w_{i,t}^{l,s}$ holds true. As explained in (Goel and Grossmann, 2006), it is enough to consider the scenario pairs that differ in only one uncertain parameter as shown in Table 1.

Table 1: Sets that differ only in the specified parameters in synthesis of process networks problem.

		Subsets of scenario pairs (M_q)			
Process	Step (l)	M_1	M_2	M_3	M_4
1	2	✓			
	3		✓		
2	2			✓	
	3				✓

3 Solution strategy

The size of the multistage stochastic MINLP model (P) increases quadratically with the number of scenarios and linearly with time periods. Therefore, it is difficult to solve this nonconvex MINLP model in fullspace for real size problems using commercial solvers. The model (P) is composed of nonconvex mixed-integer nonlinear subproblems. The subproblems are connected through initial (eq. (2.7)) and conditional (eq. (2.6)) non-anticipativity constraints. When the initial and conditional non-anticipativity constraints are relaxed, each subproblem can be solved independently. In order to take advantage of this special problem structure, we propose a duality-based branch and bound algorithm along the lines of Goel and Grossmann (2006). In the following sections, the upper and lower bounding procedures used at each node of the branch and bound tree, branching scheme, and the proposed solution algorithm are presented.

3.1 Upper bounding procedure

In the proposed algorithm, the upper bound at the root node of the branch and bound tree is found by optimizing model $\left(P_0^{LR}\right)$ in which the logic constraints (2.4)-(2.5) and disjunction (2.6) have been removed and the initial non-anticipativity constraint (2.7) are dualized as follows:

$$(P_0^{LR}) \qquad \phi_0^{LR}\left(\lambda_{d,t}^{s,s'}\right) = max \sum_{t\in T}\sum_{s\in S} p^s\ f\left(d_t^s, u_t^s\right) + \sum_{(s,s',t)\in N_I}\left[\lambda_{d,t}^{s,s'}\left(d_t^s - d_t^{s'}\right)\right] \tag{3.1}$$

$$s.t.\ (2.2)-(2.3),(2.8) \tag{3.2}$$

The parameters $\lambda_{d,t}^{s,s'}$ represent the Lagrange multipliers corresponding to constraint (2.7). In order to find the tightest upper bound generated by model $\left(P_0^{LR}\right)$ at the root node, we consider the Lagrangean dual problem $\left(P_0^{LD}\right)$ that minimizes the model $\left(P_0^{LR}\right)$ in the space of multipliers.

$$(P_0^{LD}) \qquad \phi_0^{LD} = \min_{\lambda}\ \phi_0^{LR}\left(\lambda_{d,t}^{s,s'}\right) \tag{3.3}$$

At any node n, other than the root node in the branch and bound tree, model (P_n) is optimized to calculate upper bounds. In model (P_n), not only the initial non-anticipativity constraints, but also some conditional non-anticipativity constraints are added. Initial non-anticipativity constraints are added to the model regardless of any decision whereas conditional ones from the relaxed disjunction are also included based on the branching cuts. The conditional non-anticipativity constraints that apply in node n are included in the dynamic set N_C^n. The selection of which conditional non-anticipativity constraint to include into set N_C^n, as well as some necessary cuts to be added to (P_n) will be discussed. The model (P_n), not including any such cuts, is given as follows:

$$(P_n) \qquad \phi_n = max \sum_{t\in T}\sum_{s\in S} p^s\ f\left(d_t^s, u_t^s\right) \tag{3.4}$$

s.t. (2.2)-(2.3)

$$d_t^s = d_t^{s'} \qquad \forall\left(s,s',t\right)\in N_I\cup N_C^n \tag{3.5}$$

The upper bound at node n is generated by optimizing model $\left(P_n^{LR}\right)$ in which the non-anticipativity constraint (3.5) is dualized as follows:

$$(P_n^{LR}) \qquad \phi_n^{LR} = max \sum_{t\in T}\sum_{s\in S} p^s\ f\left(d_t^s, u_t^s\right) + \sum_{(s,s',t)\in N_I\cup N_C^n}\left[\lambda_{d,t}^{s,s'}\left(d_t^s - d_t^{s'}\right)\right] \tag{3.6}$$

$$s.t.\ (2.2)-(2.3),(2.8)$$

Similar to the root node, in order to find the tightest upper bound, we again consider the Lagrangean dual problem $\left(P_n^{LD}\right)$ which is minimization of the model $\left(P_n^{LR}\right)$ in the space of multipliers,

$$(P_n^{LD}) \qquad \phi_n^{LD} = \min_{\lambda} \phi_n^{LR}\left(\lambda_{d,t}^{s,s'}\right) \tag{3.7}$$

where $\lambda_{d,t}^{s,s'}$ represents the Lagrange multipliers corresponding to constraint (3.5).

Both models $\left(P_0^{LR}\right)$ and $\left(P_n^{LR}\right)$ can be decomposed into independent nonconvex MINLP subproblems for fixed values of the multipliers. Both models are relaxations of the model (P) for any fixed values of the Lagrange multipliers, and in order to obtain valid upper bounds, these nonconvex subproblems must be globally optimized (for proof see Tarhan and Grossmann (2008)).

Minimization of the Lagrangean dual models $\left(P_0^{LD}\right)$ and $\left(P_n^{LD}\right)$ in the space of multipliers is performed by the subgradient method proposed by Fisher (1985).

3.2 Branching

In general, at any node in the branch and bound tree, the solution of the Lagrangean dual may not satisfy the dualized non-anticipativity constraint (2.7), or the conditional non-anticipativity constraints in relaxed disjunction (2.6) inferred by decisions. In this case, new branches are generated from the current node by considering the violations in the dualized non-anticipativity constraints or the relaxed disjunction.

3.2.1 Branching on the dualized non-anticipativity constraints

At any node of the branch-and-bound tree, the solution of model $\left(P_n^{LD}\right)$ may not satisfy the dualized initial or conditional non-anticipativity constraints $\left(N_I \cup N_C^n\right)$. In this case, branching is performed over the constraints in violation. If the violating non-anticipativity constraint involves discrete variables then the branching strategy divides the feasible region into two, where one of the regions is restricted to $d_t^s \leq \lfloor \tilde{d}_t \rfloor$, $d_t^{s'} \leq \lfloor \tilde{d}_t \rfloor$ and the other $d_t^s \geq \lceil \tilde{d}_t \rceil$, $d_t^{s'} \geq \lceil \tilde{d}_t \rceil$. $\tilde{d}_t$ is calculated by the probability weighted average of the variables, $\tilde{d}_t = \frac{p^s d_t^s + p^{s'} d_t^{s'}}{p^s + p^{s'}}$. In case of continuous variables, the branches are formed by $d_t^s \leq \lfloor \tilde{d}_t \rfloor - \varepsilon$, $d_t^{s'} \leq \lfloor \tilde{d}_t \rfloor - \varepsilon$ and $d_t^s \geq \lceil \tilde{d}_t \rceil + \varepsilon$, $d_t^{s'} \geq \lceil \tilde{d}_t \rceil + \varepsilon$ where $\tilde{d}_t$ is calculated as shown above.

A special case occurs during the branching of the first period non-anticipativity constraints. As these constraints should hold for each scenario pair, we can rewrite these equations as one expression, $d_1^{s_1} = d_1^{s_2} = \ldots = d_1^{s_k}$. The branching can then be performed so that the entire expression is restricted to $\leq \lfloor \tilde{d}_t \rfloor$ and to $\geq \lceil \tilde{d}_t \rceil$ (in case of continuous variables inequalities should take the forms $\leq \lfloor \tilde{d}_t \rfloor - \varepsilon$ and $\geq \lceil \tilde{d}_t \rceil + \varepsilon$.

3.2.2 Branching on the disjunctions

When the dualized non-anticipativity constraints are satisfied, it is possible to continue searching by separating the feasible space into two using the disjunction (2.6). The branching is performed using the conditional non-anticipativity constraints in the relaxed disjunctions that are supposed to hold, but do not do so given the values of the variables at the previous time periods. For instance, assume the decisions satisfy the initial non-anticipativity constraints that include the first period non-anticipativity constraints Then indistinguishable scenario pairs are inferred by using (2.4)-(2.5). Given the indistinguishable scenario pairs, if the variable values do not satisfy the conditional non-anticipativity constraints in disjunction (2.6), the branching is performed on $z_t^{s,s'}$ as shown in Figure 7.

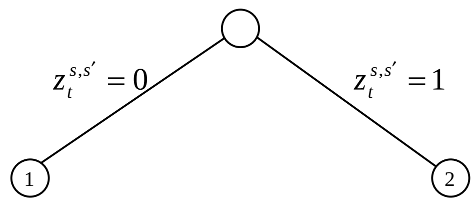

Figure 7: Branching on the disjunction in constraint (2.6).

The two generated nodes in Figure 7 (nodes 1 and 2) have different properties. At node 1, $z_t^{s,s'}$ is fixed to false/zero and at node 2, $z_t^{s,s'}$ is fixed to true/one. At node 1 due to fixing $z_t^{s,s'}$ to false, it is required to add some cuts which will guarantee the distinguishability of the scenario pair (s,s'). These cuts will be generated using constraints (2.4)-(2.5). To illustrate these cuts, we will again use the synthesis of process networks problem (Appendix A). If we fix $z_t^{s,s'}$ to false, where the scenario pair (s,s') belongs to the set M_1, then using Table 1 and (A.25) we find that logic constraints (3.8) and (3.9) must be included in scenario subproblems s and s', respectively.

$$w_{1,t}^{2,s} \Leftrightarrow False \tag{3.8}$$

$$w_{1,t}^{2,s'} \Leftrightarrow False \tag{3.9}$$

Different from node 1, at node 2 some non-anticipativity constraints coming from the relaxed disjunction are added to the set N_C^n in model (P_n) due to the fixing of $z_t^{s,s'}$ to true. Similar to node 1, cuts are generated and added to model (P_n) where $z_t^{s,s'}$ is fixed to true. These cuts are necessary to make the solution algorithm convergent. (For proof see Goel and Grossmann (2006))

3.3 Lower bounding procedure

At each node, a lower bound is found using a heuristic which converts the solution found by the upper bound procedure to a feasible solution. Usually, the solution found by the upper bound generation does not satisfy the non-anticipativity constraints. Feasible solutions are generated using a rolling horizon approach (Dimitriadis et al., 1997). Decisions in the indistinguishable scenario pairs are found by calculating the probability weighted average of variables in such scenarios. After fixing these variables and considering the resolution of uncertainty depending on the fixed decisions, the next period decisions are found iteratively by calculating similarly the probability weighted average of the variables in indistinguishable scenarios. This procedure is terminated when all scenarios are distinguishable or the end of the planning horizon is reached, whichever comes first. Then having these decisions fixed, the MINLP model is solved in full space using an outer approximation algorithm (Duran and Grossmann, 1986), which yields a feasible solution.

3.4 Solution algorithm (SP-GO)

Based on the multistage stochastic MINLP model (P) and the previous explanations about the upper and lower bounding procedures, the proposed algorithm is presented below. $\mathscr{P}$ denotes the list of open nodes each having an upper bound ϕ_n^{UB} found by the Lagrangean dual problem at node n, while ϕ^{LB} represents the objective value of the best feasible solution obtained so far. Given these variables, the major steps of the algorithm are given below.

Step 1 *Initialization:* $\phi^{LB} = -\infty$, $\phi_n^{UB} = \infty$, $\mathscr{P} = \{P_0\}$, where P_0 is the root node.

Step 2 *Termination:* If $\mathscr{P} = \emptyset$, stop. The current best solution is optimal. Otherwise, repeat steps 2 to 6.

Step 3 *Node selection:* Select and delete node n from $\mathscr{P}$ based on the best bound.

Step 4 *Bound generation:* For the root node, generate the upper bound $\left(\phi_0^{UB}\right)$ by applying a global optimization algorithm and obtain a lower bound $\left(\phi^{LB}\right)$ solutions $\left(\hat{d},\hat{x},\hat{u}\right)$ and $\left(\bar{d},\bar{x},\bar{u}\right)$, respectively. (Details of this step (steps a-g) will be explained below.)

Step 5 *Fathoming:* Delete from $\mathscr{P}$ all problems P' with $\hat{\phi}\left(P'\right) \leq \phi^{LB}$.

Step 6 *Branching:* Branch on the dualized non-anticipativity constraints or disjunctions that are violated by the solution $\left(\hat{d},\hat{x},\hat{u}\right)$ of the relaxed problem (P_n). Generate two children nodes, add them to $\mathscr{P}$.

Note that step 1 is the initialization and the algorithm iterates between steps 2 and 6 until convergence is achieved. The details of the step 4 (steps a to g) are explained below.

Step a : Set iteration $i = 0$.

Step b : While i is less than or equal to *max_iteration*, $i = i+1$, repeat steps b through h.

Step c *Generate upper bound:* For fixed multipliers, use global optimizer for each MINLP subproblem to solve $\left(P_n^{LR}\right)$ to obtain solution $\left(\hat{d},\hat{x},\hat{u}\right)$ with objective function value $\hat{\phi}$.

Step d *Update upper bound:* Update the upper bound by $\phi_n^{UB} = \min\left\{\phi_n^{UB}, \hat{\phi}\right\}$.

Step e *Generate lower bound:* Generate a feasible solution $\left(\bar{d},\bar{x},\bar{u}\right)$ with objective value $\bar{\phi}$ for the model (P) based on the solution $\left(\hat{d},\hat{x},\hat{u}\right)$ generated at step c.

Step f *Update lower bound:* Update the lower bound by $\phi^{LB} = \max\left\{\phi^{LB}, \bar{\phi}\right\}$.

Step g *Update multipliers:* Update multipliers using subgradient method.

The bottleneck of the proposed algorithm (SP-GO), is the upper bounding procedure (step c) in step 4 where every scenario subproblem has to be globally optimized in order to have valid upper bounds during the subgradient iterations. This is computationally expensive. Since it is enough to generate a valid upper bound only at the final subgradient iteration, the proposed algorithm, SP-GO, can be modified such that non-convex subproblems are solved using an outer-approximation algorithm instead of global optimization at the intermediate subgradient iterations. The reason for such a modification is that in practice during the subgradient optimization we do not intend to find the optimal multipliers, but multipliers that improve the upper bound. This modification intends to improve the Lagrange multipliers at each iteration, without solving each scenario problem globally. The drawback of the approach is not getting valid bounds during these intermediate subgradient iterations.

The following algorithm (SP-OA) combines the global optimization and outer-approximation algorithm for expediting the solution process without violating the validity of the bounds. Note that the outer-approximation algorithm can be replaced by any algorithm that assumes convexity (e.g. generalized Benders decomposition, extended cutting plane), that will find local solutions in short time.

3.5 Solution algorithm (SP-OA)

In order to incorporate this idea into the proposed algorithm (SP-GO), steps c and d need to be modified. The type of optimization (global or outer approximation) is based on the iteration i and maximum iteration limit ($max\_iteration$). If iteration i is one or the maximum iteration limit, every subproblem in the Lagrangean dual of the problem $\left(P_n^{LR}\right)$ is solved using global optimization; otherwise an outer-approximation algorithm that relies on convexity assumption is used. The reason for using global optimization at the first iteration is for initializing the variable values close to global optimal solution which is taken as an input for the outer-approximation algorithm at the second iteration. This improves the chances of finding optimal solutions during the outer-approximation algorithm. The reason for using global optimization at the last iteration is for generating valid upper bounds that will be used for pruning the parts of the branch and bound tree and calculating the duality gap. In step d, the upper bound is updated only after global optimization is used for calculating valid bounds.

Figure 8 compares the typical profiles of the Lagrangean dual as a function of multipliers. The solid lines represent a profile generated when all the subproblems are solved using global optimization, whereas the dashed lines represent a profile generated using outer approximation at the intermediate iterations. In the first and last iteration a solution is found on the profile generated by global optimization (solid lines) and during the intermediate iterations outer approximation (dashed lines) may end up in one of the local optimal solutions (x_i^{OA}).

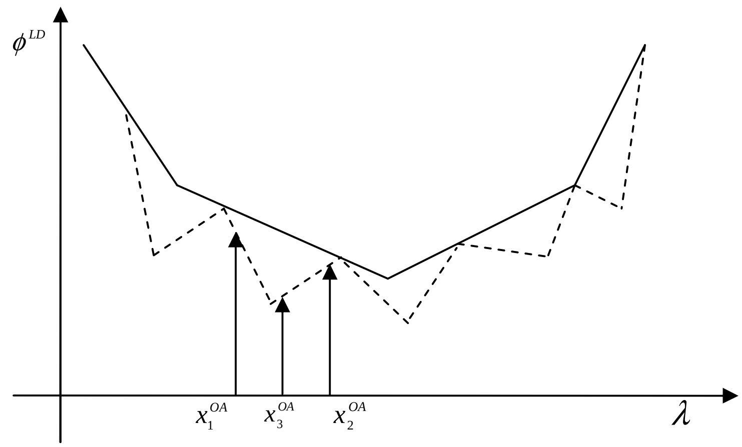

Figure 8: Graphical comparison of SP-GO and SP-OA iterations.

4 Results

In this section, we present results for comparing the solutions proposed by the algorithms SP-GO and SP-OA for the synthesis of process networks (Tarhan and Grossmann (2008)) and planning of offshore oil or gas field infrastructure problems (Tarhan et al. (2009)).

4.1 Synthesis of process networks

This section explains briefly the synthesis of process networks example and Section 4.1.1 analyzes the solution time and quality of SP-GO and SP-OA. In the synthesis of process networks problem, we consider the selection and capacity expansion of processes over a planning horizon given that there is uncertainty in the yields of some processes. Uncertainty can be reduced through investment in pilot plants. The trade-off is that pilot plants delay the introduction of processes but at a reduced uncertainty. Concave cost functions are assumed which are the sources of non-convexity in this problem.

Figure 9 shows the specific network used in the synthesis of process networks problem. The demand for the final product A over the planning horizon is known and the company must satisfy that demand. The current production takes place only in process 3, which consumes an intermediate product B from the market. In case of production shortage it is possible to buy final product A from the market at a higher cost to satisfy the demand. Inventory for both the intermediate and final product can be maintained. Two new technologies (process 1 and process 2) are available to produce the intermediate product B from two different raw materials C or D. These new technologies have uncertainty in their yields which gradually resolve over a two year period either with investments in pilot plants or with plant operation.

In Figure 9 process 3 is already operational with an existing capacity of 3000 tons/year and a known yield of 70%. The only difference between the two different technologies (Process 1 and 2) is the variance

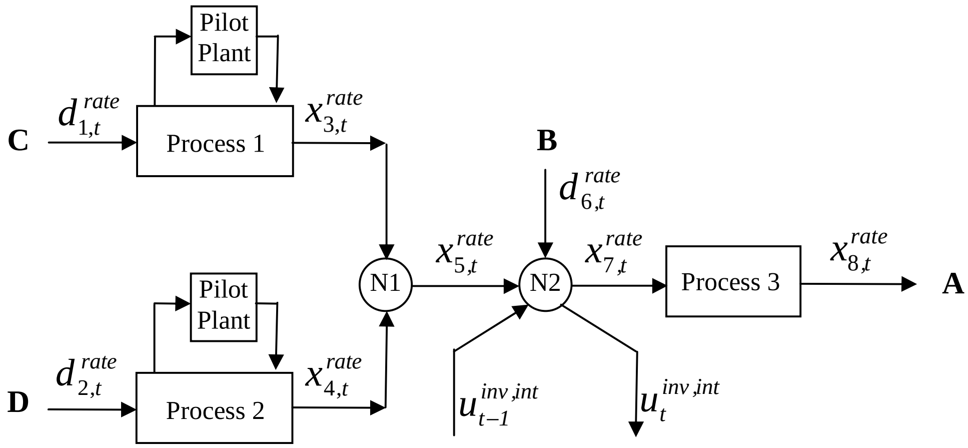

Figure 9: Schematic representation of synthesis of process networks problem.

of yield distributions. Although they possess the same mean value, 75%, process 2 has a higher variance than process 1 (see Table A.1). Process 1 can realize after first year a yield of 69 or 81% and after the second year 69, 73, 77 and 81%. Similarly, yields of process 2 after first year are 60 or 90% and after second year 60, 70, 80 and 90%. It is assumed that the probability of each scenario is the same. The detailed optimization model and the scenarios and data for this instance are presented in Appendix A.

The size of this instance in full space with 10 periods and 16 scenarios is given in Table 2. Although the problem seems to be trivial, combining 16 scenario subproblems over 10 years with initial and conditional non-anticipativity constraints, leads to a large scale problem.

Table 2: Model size in full space (16 scenarios).

	Individual Scenario	Full Space Model (16 Scenarios)
Binary Variables	160	4,912
Continuous Variables	181	2,897
Constraints	614	58,017

4.1.1 Performance analysis of SP-GO and SP-OA

The results in this section have been obtained on a Pentium-IV, 3.20 GHz Windows machine. Also, we employed AIMMS 3.8.4 for implementing the solution algorithm using solvers CPLEX 11.0, CONOPT 3.14, SNOPT 6.1, BARON 7.5.3 (Sahinidis (2000)), AOA (AIMMS Outer Approximation Module).

The specific synthesis problem has been optimized using both algorithms SP-GO and SP-OA with 2% worst case gap. Worst case gap is calculated by adding the specified gap of 1% for the global optimizer and specified gap of 1% for the branch and bound tree. The global optimization algorithms have been run with 1% optimality gap, and both algorithms were terminated when the gap in the branch and bound tree is less than 1%. The results are presented in Table 3. In this instance, although both algorithms find close solutions, SP-GO finds a slightly better (0.1%) feasible solution than SP-OA (6.636 vs. 6.629). However, while SP-GO requires 90.7 hours to complete the branch and bound search, SP-OA requires only 34.8 hours, a reduction of nearly 60%. In SP-OA the optimum solution was first found in 32.1 hours while SP-GO found it in 71.1 hours. One can also compare the best upper bounds found by the two algorithms. SP-OA reduces the best upper bound more than SP-GO by searching more nodes in branch and bound tree, but cannot find a better feasible solution.

Table 3: Performance comparison of SP-GO and SP-OA for synthesis of process networks problem.

	SP-GO	SP-OA
Lower bound ($\$ \times 10^6$) Best feasible solution	6.636	6.629
Upper bound ($\$ \times 10^6$)	6.671	6.631
Worst case gap (%)	1.52	1.02
Best feasible solution found after (hrs)	71.1	32.1
Total CPU time (hrs)	90.7	34.8

4.2 Planning of offshore oil or gas field infrastructure

In this section, we briefly describe the specific planning of offshore oil or gas field infrastructure problem. Section 4.2.1 compares the results and the solution times found by the two algorithms (SP-GO, SP-OA).

We consider a field consisting of a single reservoir (Figure 10), where a number of wells can be drilled and exploited for oil in every reservoir during the planning horizon. The problem involves making investment and operating decisions over the planning horizon. Investment decisions are selection of the number, type and capacity of facilities, and installation schedule of these facilities, as well as selection of types of wells and drilling schedule of wells. Operating decisions are amount of oil production for each time period given the limitations of the reservoirs. The goal is to capture the complex economic tradeoffs that arise from the investment and operating decisions in order to maximize the expected net present value of the project.

The details and data for the planning of oil or gas field infrastructure under uncertainty are reported extensively in Tarhan and Grossmann (2008). Briefly, the main uncertainties considered are in the initial maximum oil or gas flowrate, recoverable oil or gas volume and water breakthrough time of the reservoir, which are represented by discrete distributions. Furthermore, it is assumed that these uncertainties are not immediately realized, but are gradually revealed as a function of well drilling and production decisions. The model optimizes the investment and operating decisions over the entire planning horizon. Two instances, 1 and 2 have been optimized to show the efficiency of the proposed algorithm. In both instances the reservoir behavior is nonlinear (because of the water flowrate equations) but in instance 1, the maximum oil flowrate

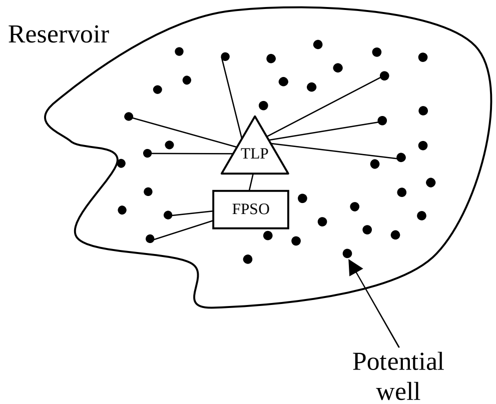

Figure 10: A Typical oil field infrastructure.

is assumed to be linear function of the cumulative oil production whereas in instance 2 it is a nonlinear curve. In both cases the nonlinearities give rise to non-convexities in the model. The size of the two instances are the same and given in Table 4.

Table 4: Model size for the oil field problems.

	Individual Scenario	Full Space Model (8 Scenarios)
Binary Variables	10	600
Integer Variables	100	800
Continuous Variables	121	969
Constraints	271	8088

4.2.1 Performance analysis of SP-GO and SP-OA

The results of this section have been obtained on a Pentium-IV, 3.20 GHz Windows machine. Also, we employed AIMMS 3.8 for implementing the solution algorithm using solvers CPLEX 10.1, CONOPT 3.14, SNOPT 6.1, BARON 7.5.3 (Sahinidis (2000)), AOA (AIMMS Outer Approximation Module).

Table 5: Performance comparison of SP-GO and SP-OA for Instances 1 and 2.

	Instance 1		Instance 2	
	SP-GO	SP-OA	SP-GO	SP-OA
Stochastic Programming Objective function value ($\$ \times 10^9$)	6.37	6.54	4.59	4.84
Total CPU time (hrs)	23	5.6	120	5.3
Worst case gap (%)	9.6	4.7	12	6.5

Table 5 shows that the solution times of both instances using SP-GO are long for practical purposes (23 and 120 hours). As presented in Table 5, in both cases using algorithm SP-OA, which combines the global optimization and outer-approximation, reduces the solution time by 76% and 96% respectively, and does not sacrifice the quality of the solution. In fact, SP-OA obtains better feasible solutions (6.54 vs. 6.37, 4.84 vs. 4.59) and narrower gaps between the upper and lower bounds than the one found with SP-GO. Similar to the results obtained in the synthesis of process networks problem, large reductions in solution time are achieved by combining local and global MINLP solvers. However, in these problems, SP-OA also found better feasible solutions.

5 Conclusion

We have presented in this paper a generic non-convex multistage MINLP model with decision dependent uncertainties. We have proposed an improvement on the duality-based branch and bound algorithm for solving the large sized instances where global optimization and outer-approximation algorithms are combined. The performance of the new algorithm (SP-OA) has been compared with the previous approach (SP-GO) for two problems, the synthesis of process networks and planning of offshore oil or gas field infrastructure. In the synthesis of process networks problem, SP-OA obtained a solution that is 0.1% worse than SP-GO while reducing the solution time about 60%. In the oil field infrastructure problem, SP-OA improved not only the best feasible solution found by SP-GO by 3-5%, but also the solution time by 85-97%. The improvement made by SP-OA can be accounted for combining the global optimizer and outer-approximation in a certain way to reduce the solution time. SP-OA takes advantage of the strengths of both algorithms, leading to large reductions in solution time without necessarily sacrificing the quality of the solution. The results also show that a combination of local and global MINLP solvers can lead to better solutions faster than using only a global solver. The global optimizer Baron was used to generate valid bounds in longer time, while the outer-approximation algorithm AOA was used to update Lagrange multipliers in a shorter time without finding the global optimum solutions.

6 Acknowledgements

The authors acknowledge the financial support from Exxon-Mobil Upstream Research Company and partial support of the National Science Foundation under grant CTS-0521769.

References

Ahmed, S., 2000. Strategic planning under uncertainty: Stochastic integer programming approaches. Phd thesis, University of Illinois at Urbana-Champaign.

Birge, J. R., Louveaux, F., 1997. Introduction to Stochastic Programming. Springer-Verlag, New York.

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_HTML/2008/10/2123.html`.

Charnes, A., Cooper, W. W., 1963. Deterministic equivalents for optimizing and satisficing under chance constraints. Operations Research 11 (1), 18–39.

Dias, M., 2002. Investment in information in petroleum, real options and revelation. In: Proceedings of the 6^{th} Annual International Conference on Real Options. Real Options Group at Cyprus, Cyprus.

Dimitriadis, A. D., Shah, N., Pantelides, C. C., 1997. RTN-based rolling horizon algorithms for medium term scheduling of multipurpose plants. Computers and Chemical Engineering 21, 1061–1066.

Duran, M. A., Grossmann, I. E., 1986. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. Mathematical Programming 36 (3), 307–339.

Fisher, M. L., 1985. An applications oriented guide to lagrangian relaxation. Interfaces 15 (2), 10–21.

Goel, V., 2005. Stochastic programming approaches for the optimal development of gas fields under uncertainty. Phd thesis, Carnegie Mellon University.

Goel, V., Grossmann, I. E., 2004. A stochastic programming approach to planning of offshore gas field developments under uncertainty in reserves. Computers and Chemical Engineering 28 (8), 1409–1429.

Goel, V., Grossmann, I. E., 2006. A class of stochastic programs with decision dependent uncertainty. Mathematical Programming 108 (2-3, Ser. B), 355–394.

Goel, V., Grossmann, I. E., El-Bakry, A. S., Mulkay, E. L., 2006. A novel branch and bound algorithm for optimal development of gas fields under uncertainty in reserves. Computers and Chemical Engineering 30, 1076–1092.

Harrison, K. W., 2007. Two-stage decision-making under uncertainty and stochasticity: Bayesian Programming. Advances in Water Resources 30 (3), 641–664.

Held, H., Woodruff, D. L., 2005. Heuristics for multi-stage interdiction of stochastic networks. Journal of Heuristics 11 (5-6), 483–500.

Jonsbraten, T., 1998. Optimization models for petroleum field exploitation. Phd thesis, Norwegian School of Economics and Business Administration.

Jonsbraten, T. W., Wets, R. J., Woodruff, D. L., 1998. A class of stochastic programs with decision dependent random elements. Annals of Operations Research 82, 83–106.

Kleywegt, A. J., Shapiro, A., Homem-de Mello, T., 2002. The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization 12 (2), 479–502.

Pflug, G. C., 1990. Online optimization of simulated markovian processes. Mathematics of Operations 15 (3), 381–395.

Ruszczyński, A., 1997. Decomposition methods in stochastic programming. Mathematical Programming 79 (1-3), 333–353.

Sahinidis, N. V., 2000. Baron: Branch-and-reduce optimization navigator. users manual. version 4.0. http://archimedes.cheme.cmu.edu/baron/manuse.pdf.

Sahinidis, N. V., 2004. Optimization under uncertainty: state-of-the-art and opportunities. Computers and Chemical Engineering 28 (6-7), 971–983.

Schultz, R., 2003. Stochastic programming with integer variables. Mathematical Programming 97 (1-2, Ser. B), 285–309.

Solak, S., 2007. Efficient solution procedures for multistage stochastic formulations of two problem classes. Phd thesis, Georgia Institute of Technology.

Stensland, G., Tjøstheim, D., 1991. Optimal decisions with reduction of uncertainty over time - an application to oil production. In: Lund, D., øksendal, B. (Eds.), Stochastic Models and Option Values. pp. 267–291.

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. A multistage stochastic programming approach for the planning of offshore oil or gas field infrastructure under decision dependent uncertainty. (To appear) Industrial and Engineering Chemistry.

Viswanath, K., Peeta, S., Salman, S. F., 2004. Investing in the links of a stochastic network to minimize expected shortest path. length. Purdue University Economics Working Papers 1167, Purdue University, Department of Economics, West Lafayette, IN.

Appendices

A Mathematical Model for Synthesis of Process Networks

Nomenclature:

Sets:

$D(s,s')$: Processes that differentiate scenarios s and s'

DK : Streams employed as decision variables $(DK \subset K)$

FK : Final product streams in the process network $(FK \subset K)$

I : Processes in the process network

IK : Streams employed as decision variables for intermediate product purchase $(IK \subset DK)$

IU : Processes with uncertain yield in the process network

$IP(i)$: Input streams to process i

$IN(j)$: Input streams to node j

J : Nodes in the process network

TJ : Nodes related to the balance of intermediate product in the process network

K : Streams in the process network

L : Levels of gradual uncertainty resolution (see Fugure 6)

$\hat{L}(i,s,s')$: Highest uncertainty resolution level in which scenarios (s,s') are indistinguishable in process $(\hat{L} \subset L)$

M_q : Scenario pairs (s,s') that differ in only one uncertain parameter

N_I : Set of initial non-anticipativity constraints

N_C : Set of conditional non-anticipativity constraints

N_C^n : Set of conditional non-anticipativity constraints at node n

$ON(j)$: Output streams from node j

$OP(i)$: Output streams from process i

Q : Subsets of the scenario pairs (s,s')

S : Possible scenarios

SK : Streams employed as state variables $(SK \subset K)$

T : Periods in time horizon

Indices:

i : Process in set I

j : Node in set J

k : Stream in set K

l : Level in set L

q : Element in set Q

s, s' : Scenario in set S

t, τ : Time period in set T

Binary variables (or equivalent Boolean variables):

$d_{i,t}^{exp,s}$: Whether or not process i is expanded in period t, scenario s

$d_{i,t}^{oper,s}$: Whether or not process i is operated in period t, scenario s

$d_{i,t}^{pilot,s}$: Whether or not pilot plant for process i is built in period t, scenario s

$w_{i,t}^{l,s}$: Whether or not the yield of process i is in level l in period t, scenario s

$z_t^{s,s'}$: Whether or not scenarios s, s' are indistinguishable in period t

Continuous variables:

$d_{i,t}^{qe,s}$: Capacity expansion of process i in period t, scenario s

$d_{k,t}^{rate,s}$: Flowrate of stream $k \in DK$ in period t, scenario s

$enpv$: Expected net present value

npv^s : Net present value of project under scenario s

$u_t^{inv-final,s}$: Amount of final product to put into inventory at the end of period t, scenario s

$u_t^{inv-int,s}$: Amount of intermediate product to put into inventory at the end of period t, scenario s

$u_t^{sales,s}$: Amount of sales of final product in period t, scenario s

$u_t^{p-final,s}$: Amount of purchases of final product in period t, scenario s

$x_{i,t}^{cap,s}$: Capacity of process i in period t, scenario s

$x_{k,t}^{rate,s}$: Flowrate of stream $k \in SK$ in period t, scenario s

Parameters:

D_t : Demand for final product in period t

$FE_{i,t}$: Fixed expansion cost for process i in period t

$FO_{i,t}$: Fixed operating cost for process i in period t

$L_{(.)}^{(.)}$: Lower bounds

p^s : Probability of scenario s

$PP_{i,t}$: Fixed and operating cost for pilot plant for process i in period t

$U^{(.)}_{(.)}$: Upper bounds

$VE_{i,t}$: Variable expansion cost for process i in period t

$VO_{k,t}$: Variable operating cost corresponding to $d^{rate,s}_{k,t}$ in period t

π^{final}_t : Purchase price for final product in period t

π^{int}_t : Purchase price for intermediate product in period t

γ^{final}_t : Cost of maintaining inventory of final product in period t

γ^{int}_t : Cost of maintaining inventory of intermediate product in period t

δ_t : Duration of period t

$\theta^{l,s}_i$: Yield of process i in level l in scenario s, $i \in IU$

θ_i : Yield of process i, $i \in I \backslash IU$

α : Exponent in the term for expansion cost $(0 < \alpha < 1)$

τ', τ'', τ''' : Time delays

In this instance, for simplicity, we assume that uncertainty resolves in two steps (i.e. in three levels, $L = \{1,2,3\}$). Based on the above definitions, the model for the synthesis of process networks is as follows: Equation (A.1) represents the expected net present value which is to be maximized over the set of scenarios S.

$$enpv = \sum_{s \in S} p^s npv^s \tag{A.1}$$

The net present value for each scenario s is given by,

$$\begin{aligned} npv^s = & -\sum_{t \in T}\sum_{i \in I}\left(FE_{i,t} d^{exp,s}_{i,t} + VE_{i,t}\left(d^{qe,s}_{i,t}\right)^{\alpha}\right) - \sum_{t \in T}\sum_{i \in I} FO_{i,t} d^{oper,s}_{i,t} \delta_t \\ & -\sum_{t \in T}\sum_{k \in DK} VO_{k,t} d^{rate,s}_{k,t} \delta_t - \sum_{t \in T}\sum_{k \in SK} VO_{k,t} x^{rate,s}_{k,t} \delta_t \\ & -\sum_{t \in T}\sum_{i \in IU} PP_{i,t} d^{pilot,s}_{i,t} - \sum_{t \in T}\left(\sum_{k \in IK} \pi^{int}_t d^{rate,s}_{k,t} + \pi^{final}_t u^{p-final,s}_t\right)\delta_t \\ & -\sum_{t \in T} \gamma^{int}_t u^{inv-int,s}_t - \sum_{t \in T} \gamma^{final}_t u^{inv-final,s}_t \qquad \forall s \in S \end{aligned} \tag{A.2}$$

where α is a fractional exponent, $0 < \alpha < 1$, which gives rise to a concave cost function.

Disjunction (A.3) represents the input-output relationships for the processes with uncertain yields, $\theta^{l,s}_i$, at each period and scenario. Note that at each time period and scenario, the yield of a process must be in one of the possible resolution levels $l \in L$.

$$\bigvee_{l \in L} \left[\begin{array}{c} w_{i,t}^{l,s} \\ \sum\limits_{k \in OP(i)} x_{k,t}^{rate,s} = \theta_i^{l,s} \left(\sum\limits_{k \in IP(i) \cap DK} d_{k,t}^{rate,s} + \sum\limits_{k \in IP(i) \cap SK} x_{k,t}^{rate,s} \right) \end{array} \right] \quad \forall i \in IU, \forall s \in S, \forall t \in T \qquad \text{(A.3)}$$

Eqs. (A.4) and (A.5) relate the Boolean variable $w_{i,t}^{l,s}$ to the previous time period decision variables so that the some set of constraints in (A.3) will be valid. Assuming that the uncertainty will resolve in two steps, there are three possibilities. If we build neither a pilot plant nor actual plant, the set of constraints in (A.3) that has zero yield (since there is no production capacity) will be valid which is expressed by (A.4). If we build/operate the actual plant one year or run pilot plant one year, the set of constraints in (A.3), that have still uncertain yields (at level 2), will be valid. This is reflected by (A.5). Finally, if the plant operates two or more years or at least one year after pilot plant operation, the set of constraints in (A.3) that have exact yield will be valid (level 3). This is captured by (A.6) since at anytime a process has to be in one of the levels of uncertainty. Note that instead of using (A.6) we could define the rule for $l = 3$ but we simplify the constraints by using (A.6). Instead of specifying a rule for level $l = 3$ as in eqs.(A.4)-(A.5), we only specify that the process yield must in one of the levels $l \in L$.

$$w_{i,t}^{l,s} \Leftrightarrow \bigwedge_{\tau=1}^{t-\tau'} \neg \left(d_{i,\tau}^{exp,s} \vee d_{i,\tau}^{pilot,s} \right) \qquad l = 1, \forall i \in IU, \forall t \in T, \forall s \in S \qquad \text{(A.4)}$$

$$w_{i,t}^{l,s} \Leftrightarrow \underline{\bigvee}_{\tau=1}^{t-\tau''} \left(d_{i,\tau}^{oper,s} \underline{\vee} d_{i,\tau}^{pilot,s} \right) \qquad l = 2, \forall i \in IU, \forall t \in T, \forall s \in S \qquad \text{(A.5)}$$

$$\underline{\bigvee}_{l \in L} w_{i,t}^{l,s} \qquad \forall i \in IU, \forall t \in T, \forall s \in S \qquad \text{(A.6)}$$

Eq. (A.7) represents the input-output relationships for the processes with certain yields, θ_i, at each period and scenario.

$$\sum_{k \in OP(i)} x_{k,t}^{rate,s} = \theta_i \left(\sum_{k \in IP(i) \cap DK} d_{k,t}^{rate,s} + \sum_{k \in IP(i) \cap SK} x_{k,t}^{rate,s} \right) \qquad \forall i \in I \backslash IU, \forall s \in S, \forall t \in T \qquad \text{(A.7)}$$

The mass balance constraints at each node in the network are given by,

$$\sum_{k \in ON(j)} x_{k,t}^{rate,s} = \sum_{k \in IN(j)} x_{k,t}^{rate,s} \qquad \forall j \in J, \forall s \in S, \forall t \in T \qquad \text{(A.8)}$$

The balance constraint that relates inventory, purchase, sales and production for final product at consecutive periods in the network is given by the following equation,

$$u_t^{inv-final,s} = u_{t-1}^{inv-final,s} + \left(\sum_{k \in FK} x_{k,t}^{rate,s} + u_t^{p-final,s} - u_t^{sales,s} \right) \delta_t \qquad \forall s \in S, \forall t \in T \qquad \text{(A.9)}$$

Similarly, the balance constraint that relates inventory, purchase, production and consumption of intermediate product at consecutive periods in the network is given by the following equation,

$$u_t^{inv-int,s} = u_{t-1}^{inv-int,s} + \left(\sum_{k \in IN(j) \cap DK} d_{k,t}^{rate,s} + \sum_{k \in IN(j) \cap SK} x_{k,t}^{rate,s} + \sum_{k \in IN(j) \cap IK} d_{k,t}^{rate,s} \right) \delta_t$$
$$\forall j \in TJ, \forall s \in S, \forall t \in T \qquad \text{(A.10)}$$

Constraint (A.11) forces the sales to satisfy demand exactly for each period t and scenario s,

$$\delta_t \, u_t^{sales,s} = D_t \qquad \forall s \in S, \forall t \in T \qquad \text{(A.11)}$$

Constraint (A.12) restricts the input flow to each process by the capacity of the corresponding process.

$$\sum_{k \in IP(i) \cap DK} d_{k,t}^{rate,s} + \sum_{k \in IP(i) \cap SK} x_{k,t}^{rate,s} \leq x_{i,t}^{cap,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad \text{(A.12)}$$

The capacity of every process in period t and scenario s is computed by the available capacity in the previous time period and the capacity expansion made (A.12).

$$x_{i,t}^{cap,s} = x_{i,t-1}^{cap,s} + d_{i,t}^{qe,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad \text{(A.13)}$$

Lower and upper bounds for capacity expansions are calculated by constraint (A.14),

$$L_i^{qe} d_{i,t}^{exp,s} \leq d_{i,t}^{qe,s} \leq U_i^{qe} d_{i,t}^{exp,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad \text{(A.14)}$$

, while constraint (A.15) restricts the output flow from each process to zero when they are not operating.

$$\sum_{k \in OP(i)} x_{k,t}^{rate,s} \leq U_i^{flow} d_{i,t}^{oper,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad \text{(A.15)}$$

Logical constraint (A.16) states that operating a process for a given scenario s at period t requires an expansion at any period τ before $t - \tau'''$, i.e. $\tau = 1, \ldots, t - \tau'''$.

$$d_{i,t}^{oper,s} \Rightarrow \bigvee_{\tau=1}^{t-\tau'''} d_{i,\tau}^{exp,s} \qquad \forall i \in IU, \forall s \in S, \forall t \in T \qquad \text{(A.16)}$$

Constraint (A.17) forces a process i to operate in period t if its capacity is expanded in that period.

$$d_{i,t}^{exp,s} \Rightarrow d_{i,t}^{oper,s} \qquad \forall i \in IU, \forall s \in S, \forall t \in T \tag{A.17}$$

Constraint (A.18) states that at most one pilot plant can be installed for each process having uncertain yield, for each scenario through the entire project life.

$$\sum_{t}^{T} d_{i,t}^{pilot,s} \leq 1 \qquad \forall i \in IU, \forall s \in S \tag{A.18}$$

The logic constraint (A.19) states that if there has been an expansion in the process i until period t in scenario s, then there is no need for a pilot plant for that process after period t.

$$\bigvee_{\tau=1}^{t} d_{i,\tau}^{exp,s} \Rightarrow \neg d_{i,t}^{pilot,s} \qquad \forall i \in IU, \forall s \in S, \forall t \in T \tag{A.19}$$

Initial non-anticipativity constraints, (A.20)-(A.24), hold regardless of the decisions made.

$$d_{i,t}^{exp,s} = d_{i,t}^{exp,s'} \qquad \forall i \in I, \forall \left(t,s,s'\right) \in N_I \tag{A.20}$$

$$d_{i,t}^{oper,s} = d_{i,t}^{oper,s'} \qquad \forall i \in I, \forall \left(t,s,s'\right) \in N_I \tag{A.21}$$

$$d_{i,t}^{pilot,s} = d_{i,t}^{pilot,s'} \qquad \forall i \in IU, \forall \left(t,s,s'\right) \in N_I \tag{A.22}$$

$$d_{i,t}^{qe,s} = d_{i,t}^{qe,s'} \qquad \forall i \in I, \forall \left(t,s,s'\right) \in N_I \tag{A.23}$$

$$d_{k,t}^{rate,s} = d_{k,t}^{rate,s'} \qquad \forall k \in DK, \forall \left(t,s,s'\right) \in N_I \tag{A.24}$$

Logic equation (A.25) states that scenarios s and s' are indistinguishable if for each process unit that differentiates the scenarios (s,s') the logic variables ($w_{i,t}^{l,s}$) are true at the level $l \in \hat{L}(i,s,s')$.

$$z_t^{s,s'} \Leftrightarrow \bigwedge_{i \in D(s,s')} \bigwedge_{l \in \hat{L}(i,s,s')} w_{i,t}^{l,s} \qquad \left(s,s'\right) \in M_q, \forall q \in Q, \forall t \in T \tag{A.25}$$

Finally, the disjunctive constraint (A.26) includes certain non-anticipativity constraints into the model if scenarios s and s' are indistinguishable at the end of period t.

$$\begin{bmatrix} z_t^{s,s'} \\ d_{i,t+1}^{oper,s} = d_{i,t+1}^{oper,s'} \quad \forall i \in I \\ d_{i,t+1}^{exp,s} = d_{i,t+1}^{exp,s'} \quad \forall i \in I \\ d_{i,t+1}^{pilot,s} = d_{i,t+1}^{pilot,s'} \quad \forall i \in IU \\ d_{i,t+1}^{QE,s} = d_{i,t+1}^{QE,s'} \quad \forall i \in I \\ d_{k,t+1}^{rate,s} = d_{k,t+1}^{rate,s'} \quad \forall k \in DK \end{bmatrix} \vee \left[\neg z_t^{s,s'}\right] \qquad \forall \left(t,s,s'\right) \in N_C \tag{A.26}$$

Data used in the instance

Note that the scenarios are generated assuming that during the partial resolution, the yield of process 1 is the lowest 0.69 or the highest 0.81. The actual yield at full resolution is either 0.69 or 0.73 if the yield is lowest at 0.69, and 0.77 or 0.81 if the yield is highest at 0.81 at partial resolution. A similar resolution approach is applied for process 2. The fractional cost exponent (α) is chosen as 0.8.

Table 6: Yield of each process in each scenario.

		Scenario															
Process	Level (l)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	2	.69	.69	.69	.69	.69	.69	.69	.69	.81	.81	.81	.81	.81	.81	.81	.81
	3	.69	.69	.69	.69	.73	.73	.73	.73	.77	.77	.77	.77	.81	.81	.81	.81
2	2	.60	.60	.90	.90	.60	.60	.90	.90	.60	.60	.90	.90	.60	.60	.90	.90
	3	.60	.70	.80	.90	.60	.70	.80	.90	.60	.70	.80	.90	.60	.70	.80	.90

Table 7: Various cost parameters used in the example.

Process	Fixed Expansion ($\$ \times 10^6$ / expansion)	Variable Expansion ($\$ \times 10^6$ / ton)	Fixed Operating ($\$ \times 10^6$ / year)	Pilot Plant ($\$ \times 10^6$/installation)
1	1.5	0.3	0.2	0.1
2	1.5	0.3	0.2	0.1
3	1.5	0.3	0.2	NA

Table 8: Capacity expansion limits for each process in the example.

Process	Max Limit $\left(U_i^{qe}\right)$ (ton $\times 10^3$ / year)	Min Limit $\left(L_i^{qe}\right)$ (ton $\times 10^3$ / year)
1	10	1
2	10	1
3	10	1

Table 9: Output flow limits from each process in the example.

Process	Max Limit $\left(U_i^{flow}\right)$ (ton $\times 10^3$ / year)	Min Limit $\left(L_i^{flow}\right)$ (ton $\times 10^3$ / year)
1	10	0.5
2	10	0.5
3	10	0.5

Table 10: Price and cost values for flows used in the example.

	A ($ $\times 10^3$ / ton)	B ($ $\times 10^3$ / ton)	C ($ $\times 10^3$ / ton)	D ($ $\times 10^3$ / ton)
Material purchase price	1	0.4	0.1	0.1
Final product sales price	0.6	NA	NA	NA
Final product inventory cost	0.1	NA	NA	NA
Intermediate product inventory cost	NA	0.1	NA	NA

Table 11: Demand for final product in the example.

	Time Period									
	1	2	3	4	5	6	7	8	9	10
Demand (ton $\times 10^3$)	1	2	4	8	8	6	6	6	8	8

Table 12: Subsets of scenario pairs specified in synthesis of process networks problem.

Subsets of scenario pairs (M_q)			
M_1	M_2	M_3	M_4
$(1,9)$	$(1,5)$	$(1,3)$	$(1,2)$
$(1,13)$	$(2,6)$	$(1,4)$	$(3,4)$
$(2,10)$	$(3,7)$	$(2,3)$	$(5,6)$
$(2,14)$	$(4,8)$	$(2,4)$	$(7,8)$
$(3,11)$	$(9,13)$	$(5,7)$	$(9,10)$
$(3,15)$	$(10,14)$	$(5,8)$	$(11,12)$
$(4,12)$	$(11,15)$	$(6,7)$	$(13,14)$
$(4,16)$	$(12,16)$	$(6,8)$	$(15,16)$
$(5,9)$		$(9,11)$	
$(5,13)$		$(9,12)$	
$(6,10)$		$(10,11)$	
$(6,14)$		$(10,12)$	
$(7,11)$		$(13,15)$	
$(7,15)$		$(13,16)$	
$(8,12)$		$(14,15)$	
$(8,16)$		$(14,16)$	