

Solution Strategies for Multistage Stochastic Programming with Endogenous Uncertainties

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

Abstract

In this paper, we present a generic Multistage Stochastic Programming (MSSP) model considering endogenous uncertainty in some of the parameters. To address the issue that the number of non-anticipativity (NA) constraints increases exponentially with the number of uncertain parameters and/or its realizations, we present a new theoretical property that significantly reduces the problem size and complements two previous properties. Since one might generate reduced models that are still too large to be solved directly, we also propose three solution strategies: a k -stage constraint strategy where we only include the NA constraints up to a specified number of stages, an iterative NAC relaxation strategy, and a Lagrangean decomposition algorithm that decomposes the problem into scenarios. Numerical results for two process network examples are presented to illustrate that the proposed solution strategies yield significant computational savings.

Keywords: multistage stochastic programming; endogenous uncertainties; non-anticipativity constraints; decomposition techniques; planning under uncertainty; process networks

1 Introduction

Stochastic programming provides a systematic framework to model problems that require decision-making in the presence of uncertainty by taking uncertainty into account of one or more parameters in terms of probability distribution functions, Birge and Louveaux (1997). This area is receiving increasing attention given the limitations of deterministic models.

*E-mail: vijaygup@andrew.cmu.edu

†To whom all correspondence should be addressed. E-mail: grossmann@cmu.edu

Jonsbraten (1998) classified uncertainty in Stochastic Programming problems into two broad categories: exogenous uncertainty (e.g. demands) where stochastic processes are independent of decisions that are taken, and endogenous uncertainty (e.g. yields) where stochastic processes are affected by these decisions. In the process systems area, Ierapetritou and Pistikopoulos (1994), Clay and Grossmann (1997) and Iyer and Grossmann (1998) solved various production planning problems that considered exogenous uncertainty and formulated as the two-stage stochastic programs. Furthermore, detailed reviews of previous work on problems with exogenous uncertainty can be found in Schultz (2003) and Sahinidis (2004). In this paper, we focus on problems with endogenous uncertainty for which there is very limited literature available.

Pflug (1990) addressed endogenous uncertainty problems in the context of discrete event dynamic systems where the underlying stochastic process depends on the optimization decisions. Jonsbraten et al. (1998) proposed an implicit enumeration algorithm for the problems in this class where decisions that affect the uncertain parameter values are made at the first stage. Ahmed (2000) presented several examples having decision dependent uncertainties that were formulated as MILP problems and solved by LP-based branch & bound algorithms. Moreover, Viswanath et al. (2004) and Held and Woodruff (2005) addressed the endogenous uncertainty problems where decisions can alter the probability distributions.

Recently, few practical applications that involve multistage stochastic programming with endogenous uncertainty have been addressed. Goel and Grossmann (2004) and Goel et al. (2006) dealt with the gas field development problem under uncertainty in size and quality of reserves where decisions on the timing of field drilling yield an immediate resolution of the uncertainty. These authors proposed a branch and bound solution method based on a Lagrangean decomposition scheme that relies on relaxing the disjunctions and logic constraints for the non-anticipativity constraints. Later, Tarhan et al. (2009) extended their approach to gradual resolution of endogenous uncertainty and included nonlinearities in the reservoir model. Tarhan and Grossmann (2008) also applied endogenous uncertainty in the synthesis of process networks problem with uncertain yields and used gradual uncertainty resolution in the model. Solak (2007) considered the project portfolio optimization problem that deals with the selection of research and development projects and determination of optimal resource allocations under decision dependent uncertainty where uncertainty resolved gradually. The author used the sample average

approximation method for solving the problem, where the sample problems were solved through Lagrangean relaxation and heuristics. Boland et al. (2008) addressed the open pit mine production scheduling problem considering endogenous uncertainty in the total amount of rock and metal contained in it, where the excavation decisions resolve this uncertainty. These authors also compared the fullspace results for this mine-scheduling problem with the one where non-anticipativity constraints were included as the ‘lazy constraints’ during the solution. Colvin and Maravelias (2008, 2010) presented several theoretical properties, specifically for the problem of scheduling of clinical trials having uncertain outcomes in the pharmaceutical R&D pipeline, and developed a branch and cut framework to solve these MSSP problems with endogenous uncertainty under the assumption that only few non-anticipativity constraints be active at the optimal solution.

The outline of this paper is as follows. Firstly, we present a brief background on stochastic programming. Next, we introduce the problem statement and the multistage stochastic programming model for endogenous uncertainty problems. Three theoretical properties are then identified for the model and used to formulate a reduced model. Then, a k -stage constraint solution approach and NAC relaxation strategy are proposed using the reduced model to solve large instance of these problems. Finally, a Lagrangean decomposition algorithm for multistage stochastic programming problems with endogenous uncertainty is presented that provides good bounds at the root node. Numerical results of two process network problems are presented for the various proposed solution approaches.

2 Background

A Stochastic Program is a mathematical program in which some of the parameters defining a problem instance are random (e.g. demand, yield). The basic idea behind stochastic programming is to make some decisions now (stage 1) and to take some corrective action (recourse) in the future, after revelation of the uncertainty (stages 2,3,...). If there are only two stages then the problem corresponds to a 2-stage stochastic program, while in a multistage stochastic program the uncertainty is revealed sequentially, i.e. in multiple stages (time periods), and the decision-maker can take corrective action over a sequence of stages. In the two-stage and multistage case the cost of the decisions and the expected cost of the recourse actions are optimized.

We assume in this paper that the uncertain parameters follow discrete probability distributions and that the planning horizon consists of a fixed number of time periods that correspond to decision points. Using these two assumptions, the stochastic process can be represented with scenario trees. In a scenario tree (Figure 1-a) each node represents a possible state of the system at a given time period. Each arc represents the possible transition from one state in time period t to another state in time period $t+1$, where each state is associated with the probabilistic outcome of a given uncertain parameter. A path from the root node to a leaf node represents a scenario. An alternative representation of the scenario tree was proposed by Ruszczyński (1997) where each scenario is represented by a set of unique nodes (Figure 1-b). The horizontal lines connecting nodes in time period t mean that nodes are identical as they have the same information and those scenarios are said to be indistinguishable in that time period. These horizontal lines correspond to the non-anticipativity (NA) constraints in the model that link different scenarios and prevent the problem from being decomposable.

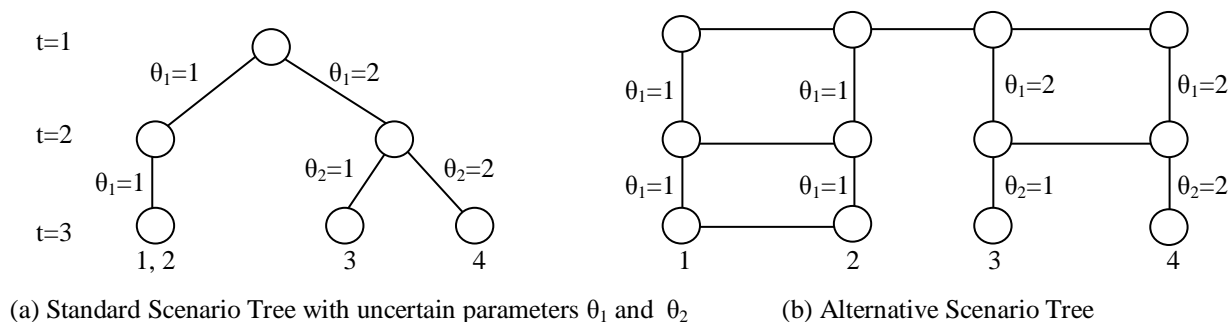


Figure 1: Tree representations for discrete uncertainties over 3 stages.

In this paper, we focus on multistage stochastic programming (MSSP) problems with endogenous uncertainty where the structure of scenario tree is decision-dependent. Therefore, we use the above alternative scenario tree representation to model these problems effectively.

3 Problem Statement

In the class of problems under consideration, the time horizon is represented by the discrete set of time periods $T = \{1, 2, \dots, T\}$. Set $I = \{1, 2, \dots, I\}$ represents the set of “sources” of endogenous uncertainty, while θ_i represents the endogenous uncertain parameter associated with source $i \in I$. The discrete set of possible realizations for θ_i is represented by Φ_i . The resolution of uncertainty in θ_i depends on the binary decision variables $b_{i,t}$. Specifically, the uncertainty in θ_i

will be resolved in time period t if binary decision $b_{i,t} = 1$ and $b_{i,\tau} = 0, \forall \tau < t$. Note that the parameters θ_i represent intrinsic properties of source i and are independent of time period. Besides the decisions represented by variables $b_{i,t}$, other decisions to be made in time period t are represented by variables y_t and x_t where these are decisions made at the beginning and end of the corresponding time period t .

The sequence of events in each time period is as follows. Decisions y_t and $b_{i,t}$ are implemented at the beginning of time period t . This is followed by the resolution of uncertainty in the endogenous parameter θ_i for source i if $b_{i,t} = 1$ and $b_{i,\tau} = 0 \forall \tau < t$. The state variables (w_t) are calculated based on the decision variables that are selected, while the recourse variables (x_t) are decisions implemented at the end of each period.

In general, the variables $b_{i,t}$ may represent investment decisions associated with source i . In the gas field problem considered by Goel et al. (2006), these variables represent whether or not investment is made at field i in time period t . The uncertainty associated with a field is resolved in time period t only if investment is carried out at that field in time period t , while no investments have been made at that field in the past. Similarly, for capacity expansion planning problems these decisions represent whether or not unit i is installed in time period t . However, in this case we assume that the uncertainty associated with a process gets resolved as soon as initial investment is made in that process. Note that for ease of exposition, we assume that there is only one endogenous uncertain parameter associated with source i for all $i \in I$. Thus, θ_i is a scalar for all $i \in I$. Moreover, the problem statement presented here is the specific case of the one that is described in Goel and Grossmann (2006).

4 Model

The Multistage Stochastic Programming model (MSSP⁰) with endogenous uncertainty can be represented as a mixed-integer disjunctive programming model as described in Goel and Grossmann (2006).

$$\begin{aligned}
 \text{(MSSP}^0\text{)} \quad & \min \sum_{s \in S} p^s \sum_{t \in T} \left(c_t^{ws} w_t^s + c_t^{xs} x_t^s + c_t^{ys} y_t^s + \sum_{i \in I} c_{i,t}^{bs} b_{i,t}^s \right) & (1) \\
 \text{s.t.} \quad & \sum_{\substack{\tau \in T, \\ \tau \leq t}} \left(A_{\tau,t}^{ws} w_\tau^s + A_{\tau,t}^{xs} x_\tau^s + A_{\tau,t}^{ys} y_\tau^s + \sum_{i \in I} A_{i,\tau,t}^{bs} b_{i,\tau}^s \right) \leq a_t^s \quad \forall s \in S, \forall t \in T & (2)
 \end{aligned}$$

$$b_{i,1}^s = b_{i,1}^{s'} \quad \forall s, s' \in S, \forall i \in I, s \neq s' \quad (3a)$$

$$y_1^s = y_1^{s'} \quad \forall s, s' \in S, s \neq s' \quad (3b)$$

$$Z_t^{s,s'} \Leftrightarrow \bigwedge_{i \in D(s,s')} \left[\bigwedge_{\tau=1}^t \neg(b_{i,\tau}^s) \right] \quad \forall s, s' \in S, \forall t \in T, s \neq s' \quad (4)$$

$$\left[\begin{array}{c} Z_t^{s,s'} \\ x_t^s = x_t^{s'} \\ b_{i,t+1}^s = b_{i,t+1}^{s'} \quad \forall i \in I \\ y_{t+1}^s = y_{t+1}^{s'} \end{array} \right] \vee [\neg Z_t^{s,s'}] \quad \forall s, s' \in S, \forall t \in T, s \neq s' \quad (5)$$

$$w_t^s \in W_t^s, x_t^s \in X_t^s, y_t^s \in Y_t^s, b_{i,t}^s \in \{0,1\} \quad \forall s \in S, \forall t \in T, \forall i \in I$$

$$Z_t^{s,s'} \in \{True, False\} \quad \forall s, s' \in S, \forall t \in T$$

The objective function (1) in the above model (MSSP⁰) minimizes the expectation of an economic criterion. For a particular scenario, inequality (2) represents constraints that govern decisions in time period t and link decisions across time periods. First time period non-anticipativity (NA) constraints are given by equations (3a) and (3b), while conditional NA constraints that are written for the later time periods in terms of decisions $b_{i,t}^s$ are given by (4) and (5). Note that the set $D(s, s')$ that is used in the equation (4) is defined as follows:

$$D(s, s') = \{i \mid i \in I, \theta_i^s \neq \theta_i^{s'}\}$$

The idea of non-anticipativity is that the decisions at time t can only be affected by the decisions $(y_t^s, b_{i,t}^s)$ made before time period t . These constraints state that if two scenarios are indistinguishable in time period t (i.e. they are the same), then decisions for s and s' in t should be the same. It should also be noted that problem (MSSP⁰) can be reformulated as an MILP as described in Goel and Grossmann (2006) by replacing the equations (4) and (5) with integer and mixed-integer constraints, respectively.

5 Model Reduction Scheme

NA constraints like the ones in (3a), (3b) and (5) are essential in multi-stage stochastic programming to ensure that our current decisions do not anticipate future outcomes. When the model (MSSP⁰) is reformulated as an MILP problem, the difficulty is that the NA constraints typically represent around 80% of the total constraints and grow quadratically in the number of

scenarios, making real-world size problems intractable. To overcome this limitation, we present three theoretical properties that allow us to formulate significantly reduced MSSP models.

Let us assume that there are p uncertain parameters $(\theta_1, \theta_2, \theta_3, \dots, \theta_p)$ each of which has k realizations $(\ell_1, \ell_2, \ell_3, \dots, \ell_k)$. Then the total number of combinations of realizations of these parameters will be k^p each of which will define a scenario s . For these $S = k^p$ scenarios there will be a total of $S(S - 1)$ scenario pairs (s, s') each of which corresponds to a NA constraint. The following properties significantly reduce the problem size by reducing the number of these scenario pairs (s, s') and the corresponding NA constraints. The first two properties were proposed by Goel and Grossmann (2006).

Property 1. *If scenario pair (s, s') is indistinguishable at stage t , so is (s', s) . Therefore, we have to consider only one of these scenario pairs (i. e. (s, s') such that $s < s'$).*

Proof. See Goel and Grossmann (2006).

Property 2. *It is sufficient to express NA constraints for the pairs of scenarios (s, s') that differ in the outcome of only one uncertain parameter.*

Proof. See Goel and Grossmann (2006).

Property 1 is based on the symmetry of the scenario pairs (s, s') and prevents duplication of the NA constraints for the same pair of scenarios (s, s') in the model. On the other hand, Property 2 exploits the fact that the NA constraints between those scenarios which differ in the realizations of more than one uncertain parameter is implicitly enforced by considering the NA constraints for the one that differ in realization of only one uncertain parameter. Therefore, it is sufficient to include a subset of scenario pairs corresponding to those that differ in realization of one uncertain parameter. Properties 1 and 2 are further illustrated by a small example in the next section. Although, these two properties significantly reduce the number of scenario pairs for the NA constraints, there are still many of these scenario pairs that are connected implicitly and that can be removed. This motivates us to find these scenario pairs systematically to further reduce the size of the problem and establish a new Property 3.

Property 3 basically exploits transitivity relationship among scenario pairs (s, s') and is an extension of Property 2 to those cases where uncertain parameters have more than two realizations. In that case, according to the Property 2 all the scenario pairs that differ in just one uncertain parameter will be included in the model for the NA constraints and there will be multiple links among those scenarios pairs that corresponds to a single uncertain parameter.

Some of these multiple links among scenarios are not needed because of the fact that many of these scenarios that corresponds to a single uncertain parameter are such that they can be only realized at the same time irrespective of any decisions taken during the planning horizon and hence, we can take advantage of the transitivity relation among these scenarios.

Therefore, the new property establishes that for an endogenous uncertainty problem, it is sufficient to express NA constraints uniquely only for those scenario pairs (s, s') for which the corresponding scenarios exist in the same scenario set L_{pm} defined below. The scenarios within each set L_{pm} can only be realized at the same time irrespective of the other realizations during the given time horizon and differ only in one unique uncertain parameter θ_p . Specifically, Property 3 can be stated as:

Property 3. *For an endogenous uncertainty problem having p uncertain parameters and S scenarios, the maximum number of scenario pairs (s, s') required to represent the non-anticipativity are $p(S - S^{p-1/p})$.*

Proof. Suppose that for an endogenous uncertainty problem,

p is the number of uncertain parameters = $(\theta_1, \theta_2, \theta_3, \dots, \theta_p)$

k is the number of realizations of each uncertain parameter = $(\ell_1, \ell_2, \ell_3, \dots, \ell_k)$

Therefore, the total number of scenarios are $S = k^p$

For each uncertain parameter θ_p , there are k^{p-1} number of scenario sets L_{pm} such that,

$$L_{pm} = (s_m, s_{k^{p-1}+m}, s_{2k^{p-1}+m}, \dots, s_{(k-1)k^{p-1}+m}) \quad p = 1, 2, \dots, p \quad m = 1, 2, \dots, k^{p-1}$$

Hence, each of these L_{pm} scenario sets has k scenario elements and an index for these scenario elements as defined above. The characteristic of these k scenarios within a set L_{pm} is that uncertainty in these scenario elements within a set L_{pm} is realized at the same time irrespective of the other realization during the specified time horizon, and these scenarios elements in a set have the same realizations for all the uncertain parameters except for that particular uncertain parameter θ_p for which set L_{pm} is written.

In other words, the k scenarios in a set L_{pm} differ in the realization of the uncertain parameter θ_p only and realized at the same time irrespective of other realizations. There is no

other set with these properties. Also, according to Property 2, it is sufficient to express NA constraint for those scenario pairs that differ in the realization of only one uncertain parameter. Therefore, we do not need to include scenario pairs (s, s') that differ in realization of more than one uncertain parameter.

Moreover, as the uncertainty in these k scenarios in a set L_{pm} is realized at same time, it is sufficient to express non-anticipativity uniquely in these k scenarios only. Hence, $k-1$ scenario pairs (s, s') will be required to link k scenario elements in each of L_{pm} sets, i.e. $k-1$ equations are required to represent non-anticipativity for each of these k^{p-1} sets for a particular uncertain parameter θ_p . Therefore, the total number of scenario pairs (s, s') required for non-anticipativity are $p(S - S^{p-1/p})$ or $pk^{p-1}(k-1)$. \square

The proposed Property 3 can be used in addition to earlier Properties 1 and 2 to reduce the model size as explained in the next section with a small example.

6 Reduced Model Formulation

In this section we apply the three properties described above in order to reduce the size of the model (MSSP⁰). Let us define,

P : Set of scenario pairs (s, s') for NA constraint in the model (MSSP⁰)

P_1 : Set of scenario pairs (s, s') for NA constraints after applying Property 1

P_2 : Set of scenario pairs (s, s') for NA constraints after applying Properties 1 and 2

P_3 : Set of scenario pairs (s, s') for NA constraints after applying Properties 1, 2 and 3

The relation between these sets can be stated as, $P_3 \subseteq P_2 \subseteq P_1 \subseteq P$.

The reduced model (MSSP^R) that is formulated from the original model (MSSP⁰) by considering NA constraints for scenario pairs (s, s') within the set P_3 for the equations (3a), (3b), (4) and (5) is given as follows:

$$\text{(MSSP}^{\text{R}}\text{)} \quad \min \sum_{s \in S} p^s \sum_{t \in T} \left(c_t^{ws} w_t^s + c_t^{xs} x_t^s + c_t^{ys} y_t^s + \sum_{i \in I} c_{i,t}^{bs} b_{i,t}^s \right) \quad (1)$$

$$s.t. \sum_{\substack{\tau \in T, \\ \tau \leq t}} \left(A_{\tau,t}^{ws} w_{\tau}^s + A_{\tau,t}^{xs} x_{\tau}^s + A_{\tau,t}^{ys} y_{\tau}^s + \sum_{i \in I} A_{i,\tau,t}^{bs} b_{i,\tau}^s \right) \leq a_t^s \quad \forall s \in S, \forall t \in T \quad (2)$$

$$b_{i,1}^s = b_{i,1}^{s'} \quad \forall (s, s') \in P_3, \forall i \in I \quad (6a)$$

$$y_1^s = y_1^{s'} \quad \forall (s, s') \in P_3 \quad (6b)$$

$$Z_t^{s,s'} \Leftrightarrow \bigwedge_{i \in D(s,s')} \left[\bigwedge_{\tau=1}^t \neg(b_{i,\tau}^s) \right] \quad \forall (s, s') \in P_3, \forall t \in T \quad (7)$$

$$\left[\begin{array}{c} Z_t^{s,s'} \\ x_t^s = x_t^{s'} \\ b_{i,t+1}^s = b_{i,t+1}^{s'} \quad \forall i \in I \\ y_{t+1}^s = y_{t+1}^{s'} \end{array} \right] \vee [\neg Z_t^{s,s'}] \quad \forall (s, s') \in P_3, \forall t \in T \quad (8)$$

$$w_t^s \in W_t^s, x_t^s \in X_t^s, y_t^s \in Y_t^s, b_{i,t}^s \in \{0,1\} \quad \forall s \in S, \forall t \in T, \forall i \in I$$

$$Z_t^{s,s'} \in \{True, False\} \quad \forall s, s' \in S, \forall t \in T$$

Theorem 1. *The optimum solution of the Reduced model (MSSP^R) is the same as the optimum solution of the Original Model (MSSP⁰).*

The proof follows trivially from applying Properties 1-3. To illustrate the effect of the proposed properties on the problem size, we consider a case of endogenous uncertainty problem having 2 uncertain parameters, *i.e.* (θ_1, θ_2) . Each of these uncertain parameters has three realizations (ℓ_1, ℓ_2, ℓ_3) which give rise to a total of 9 scenarios shown in Table 1.

Table 1: 9 Scenarios for the given example

Scenario (s)	1	2	3	4	5	6	7	8	9
θ_1	ℓ_1	ℓ_2	ℓ_3	ℓ_1	ℓ_2	ℓ_3	ℓ_1	ℓ_2	ℓ_3
θ_2	ℓ_1	ℓ_1	ℓ_1	ℓ_2	ℓ_2	ℓ_2	ℓ_3	ℓ_3	ℓ_3

According to the original model (MSSP⁰), a total of 72 scenario pairs will be required to represent non-anticipativity in the above problem as shown in Table 2(a) where each element in the table represent the indices of uncertain parameters, (θ_1, θ_2) that differentiate the corresponding scenarios s and s' , *i.e.* set $D(s, s')$.

However, if we use Property 1 (*i.e.* (s, s') such that $s < s'$) the number of scenario pairs reduces to 36 from 72 due to the symmetry of the scenario pairs as seen in Table 2(b). Now, if we apply Property 2 (*i.e.* consider the scenario pairs which differ in realization of only one uncertain parameter) then (s, s') becomes 18 by removing those scenario pairs have more than one element in the set $D(s, s')$ as seen in Table 2(c). But out of these 18 scenario pairs, only 12 are sufficient as seen in Table 2(d) to uniquely define the non-anticipativity that also satisfies the requirement of Property 3. This is due to the transitivity relation among the scenarios pairs corresponding to a single uncertain parameter and their characteristic of being realized at the

same time irrespective of the other decisions as explained in the previous section. Hence, there is 83.33% reduction (i.e. from 72 to 12) in the scenario pairs (or problem size) on using the three theoretical properties. Note that for this example $L_{11} = (s_1, s_2, s_3)$, $L_{12} = (s_4, s_5, s_6)$, $L_{13} = (s_7, s_8, s_9)$, $L_{21} = (s_1, s_4, s_7)$, $L_{22} = (s_2, s_5, s_8)$ and $L_{23} = (s_3, s_6, s_9)$ according to the definition of these sets described earlier and each of these set corresponds to two scenario pairs among three scenario elements it contains giving a total of 12 scenario pairs.

Table 2: Scenario pairs and corresponding differentiating set $D(s, s')$ for the 9 scenario example

$D(s,s')$	1	2	3	4	5	6	7	8	9	$D(s,s')$	1	2	3	4	5	6	7	8	9	
1		1	1	2	1,2	1,2	2	1,2	1,2	1		1	1	2				2		
2	1		1	1,2	2	1,2	1,2	2	1,2	2			1		2				2	
3	1	1		1,2	1,2	2	1,2	1,2	2	3						2				2
4	2	1,2	1,2		1	1	2	1,2	1,2	4					1	1	2			
5	1,2	2	1,2	1		1	1,2	2	1,2	5						1			2	
6	1,2	1,2	2	1	1		1,2	1,2	2	6										2
7	2	1,2	1,2	2	1,2	1,2		1	1	7								1	1	
8	1,2	2	1,2	1,2	2	1,2	1		1	8										1
9	1,2	1,2	2	1,2	1,2	2	1	1		9										

(a) 72 Scenario pairs in the original model (MSSP⁰)

(c) 18 Scenario pairs after using Properties 1-2

$D(s,s')$	1	2	3	4	5	6	7	8	9	$D(s,s')$	1	2	3	4	5	6	7	8	9	
1		1	1	2	1,2	1,2	2	1,2	1,2	1		1		2						
2			1	1,2	2	1,2	1,2	2	1,2	2			1		2					
3				1,2	1,2	2	1,2	1,2	2	3						2				
4					1	1	2	1,2	1,2	4					1		2			
5						1	1,2	2	1,2	5						1			2	
6							1,2	1,2	2	6										2
7								1	1	7								1		
8									1	8										1
9										9										

(b) 36 Scenario pairs after using Property 1

(d) 12 Scenario pairs after using Properties 1-3

The graphical illustration of the model reduction scheme for the above 9 scenario example can be seen in Figure 2. Property 1 basically removes one of the two links between scenarios 1 and 2 in the figure. Scenarios 1 and 5 differ in both the uncertain parameters and due to the implicit connection between these scenarios through links 1-2 and 2-5 each of which corresponds to a single uncertain parameter, Property 2 can be used to remove the link 1-5. Because scenarios 1, 4 and 7 differ in the realization of just the second uncertain parameter θ_2

and can only be realized simultaneously, they can be expressed by unique link among them. Therefore Property 3 removes the link 1-7 and still allowing scenarios 1 and 7 to take non-anticipative decisions through the links 1-4 and 4-7. The other similar links removed by these properties are not shown in the figure for clarity.

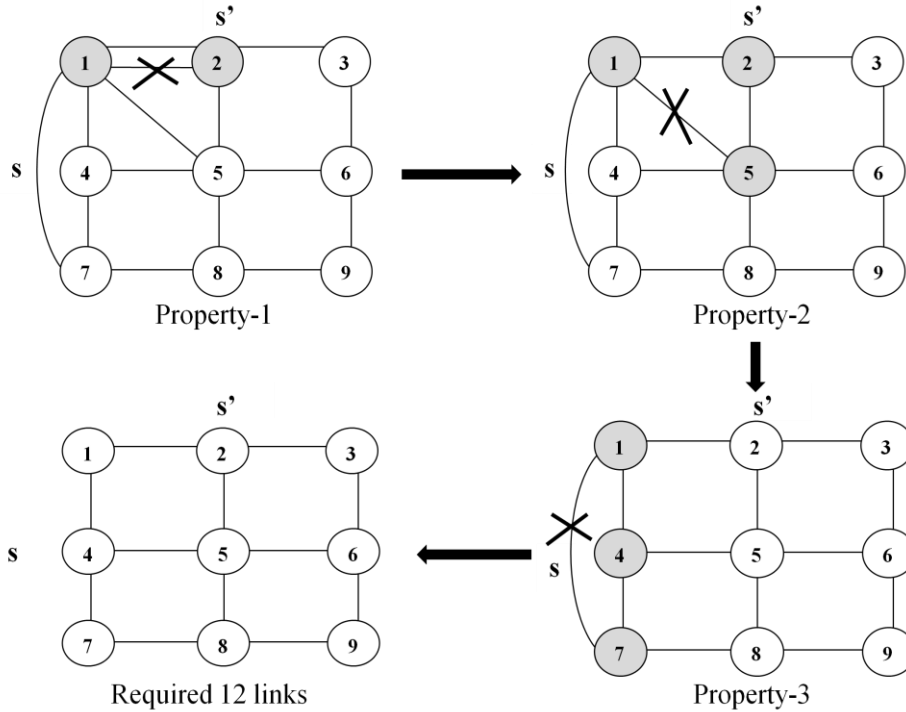


Figure 2: Model Reduction Scheme for 9 scenario example

Note that the number of scenario pairs using Properties 1-3 will be smaller compared to using Properties 1-2 only if the number of realization of uncertain parameters is more than two. Otherwise we will get the same number of scenarios in both the cases. Therefore, in contrast to the earlier properties by Goel and Grossmann (2006), the proposed Property 3 can be regarded as the extension of the Property 2 to the cases where uncertain parameters have more than two realizations. Moreover, the effect of these properties on the problem size and solution time becomes very significant for the problems having large number of scenarios and/or having many realizations of each uncertain parameter.

7 Solution Strategies

Although the model formulation in the previous section greatly reduces the size of the multistage stochastic programs with endogenous uncertainties, given the exponential increase in the problem size with the number of uncertain parameters and its realizations, these problems may

not be solvable in reasonable computational time. Hence, we may need some special solution techniques to solve large-scale problems in this class as discussed in this section.

7.1 k -stage Constraint Strategy

We know that NA constraints play a major role in the size of any Multistage Stochastic program and most of them are inactive at the optimum solution of the problem, particularly in the later time periods. This observation motivates us to include only the subset of these constraints, corresponding up to the first k -stages of the problem which are assumed to be critical for defining the optimum solution of the problem. By defining ST as the set of k initial stages for which NA constraints are to be included, the proposed k -stage constraint formulation that is obtained from the reduced model by modifying the set T in equations (7) and (8) is as follows:

$$(MSSP^{SC}) \quad \min \sum_{s \in S} p^s \sum_{t \in T} \left(c_t^{ws} w_t^s + c_t^{xs} x_t^s + c_t^{ys} y_t^s + \sum_{i \in I} c_{i,t}^{bs} b_{i,t}^s \right) \quad (1)$$

$$s.t. \sum_{\substack{\tau \in T, \\ \tau \leq t}} \left(A_{\tau,t}^{ws} w_\tau^s + A_{\tau,t}^{xs} x_\tau^s + A_{\tau,t}^{ys} y_\tau^s + \sum_{i \in I} A_{i,\tau,t}^{bs} b_{i,\tau}^s \right) \leq a_t^s \quad \forall s \in S, \forall t \in T \quad (2)$$

$$b_{i,1}^s = b_{i,1}^{s'} \quad \forall (s, s') \in P_3, \forall i \in I \quad (6a)$$

$$y_1^s = y_1^{s'} \quad \forall (s, s') \in P_3 \quad (6b)$$

$$Z_t^{s,s'} \Leftrightarrow \bigwedge_{i \in D(s,s')} \left[\bigwedge_{\tau=1}^t \neg (b_{i,\tau}^s) \right] \quad \forall (s, s') \in P_3, \forall t \in ST \quad (9)$$

$$\left[\begin{array}{c} Z_t^{s,s'} \\ x_t^s = x_t^{s'} \\ b_{i,t+1}^s = b_{i,t+1}^{s'} \quad \forall i \in I \\ y_{t+1}^s = y_{t+1}^{s'} \end{array} \right] \vee \left[\neg Z_t^{s,s'} \right] \quad \forall (s, s') \in P_3, \forall t \in ST \quad (10)$$

$$w_t^s \in W_t^s, x_t^s \in X_t^s, y_t^s \in Y_t^s, b_{i,t}^s \in \{0,1\} \quad \forall s \in S, \forall t \in T, \forall i \in I$$

$$Z_t^{s,s'} \in \{True, False\} \quad \forall s, s' \in S, \forall t \in T$$

The above model can be solved successively by starting with a fixed number of stages (say $k=2$) with NA constraints and increasing the number of stages, i.e. k value, if NA constraints of those stages greater than k are violated. The following two propositions are established to implement the proposed k -stage constraint strategy:

Proposition 1. *The k -stage constraint model ($MSSP^{SC}$) provides a valid lower bound on the Original Model ($MSSP^0$) and the Reduced Model ($MSSP^R$).*

Proof. It can be seen from the Reduced Model ($MSSP^R$) and the stage constraint model ($MSSP^{SC}$) that they are identical except the constraints that corresponds to the conditional NA constraints. More specifically, equations (9) and (10) are written for the subset of stages ST instead of all the stages T in equations (7) and (8) respectively. Therefore, the k -stage constraint model ($MSSP^{SC}$) can be regarded as the relaxation of the Reduced Model ($MSSP^R$) where we neglect the conditional NA constraints for the stages that are not the elements of the set ST. Hence, the k -stage constraint model ($MSSP^{SC}$) provides a valid lower bound on the Reduced Model ($MSSP^R$). As models ($MSSP^R$) and ($MSSP^0$) are equivalent, the k -stage constraint model ($MSSP^{SC}$) also provides a valid lower bound on the Original Model ($MSSP^0$). \square

Proposition 2. *The k -stage constraint model ($MSSP^{SC}$) provides the optimum solution to the Original Model ($MSSP^0$) and the Reduced Model ($MSSP^R$), if there is no realization of any of the endogenous uncertain parameter after specified stages in the solution that is obtained.*

Proof. The proof follows from the fact that if there is no realization of any of the uncertain parameter after specified stage k in the solution, then there will be no new information available to any scenario from period $k + 1$ to end of the planning horizon T . Therefore, the state of the system corresponding to each scenario will be the same from period k to T . Moreover, the scenario pairs that have already being distinguished within the first k stages according to the logic condition of the non-anticipativity, there will not be any need to include NA constraints for these scenario pairs. On the other hand, if there are some scenario pairs that have not been distinguished until stage k , and as there in no further realization of uncertainty, these scenarios will have the same information from period 1 to T and will have the same decisions. Hence, the NA constraints from period $k + 1$ to T will automatically be satisfied for these scenario pairs. Given that the reduced model ($MSSP^R$) and the stage constraint model ($MSSP^{SC}$) are identical except the conditional NA constraints that were relaxed, i.e. from period $k + 1$ to T in the stage constraint model, and because the NA from period $k + 1$ to T are satisfied in the solution of k stage constraint model if there is no realization of uncertain parameter after stage k as discussed earlier, the solution of the stage constraint model corresponding to the current stage k will be the optimum solution for the reduced model ($MSSP^R$). As models ($MSSP^R$) and ($MSSP^0$) are equivalent, the k -stage constraint model ($MSSP^{SC}$) also provides an optimal solution to the

Original Model (MSSP⁰) if the above condition is satisfied. □

The step-by-step procedure to implement the proposed k -stage constraint strategy is as follows:

Step 1: Set the effective number of stages k (usually $k=2$) and lower bound = $-\infty$.

Step 2: Include NA constraints for the specified number of stages k in the model (MSSP^{SC}) and solve.

Step 3: If Proposition 2 is satisfied, i.e. there is no realization of any of the uncertain parameter after the current stage k , Stop. Optimal solution is found; else go to Step 4.

Step 4: If Proposition 2 is not satisfied, update the lower bound using the solution of the model (MSSP^{SC}) for the specified value of k . Set $k=k+1$ and go to the Step 2.

The following remarks can be made about the proposed k -stage constraint strategy:

1. There are two cases involved while checking the satisfaction of Proposition 2 in step 3 of the above procedure. In the first case if there is neither investment nor expansion decision in the later stages in the solution, then we can ensure that the Proposition 2 is satisfied and solution is optimal by just examining the solution by inspection. In case that there are expansions in the later stages and no new investment, then the NA constraints corresponding to the later stages are also satisfied, i.e. Proposition 2 holds true and the solution is optimal.
2. The lower bounds obtained from the above procedure are generally very tight and the corresponding solution is very close to the feasible solution to the original problem. Therefore, this solution can be used to obtain a good feasible solution, i.e. upper bound, and one can also evaluate the quality of the solution that is obtained.
3. In case that the iterations during the above solution procedure are computationally expensive, one can use the solution of the previous iteration to determine a good value of k that can be used in the next iteration to fix the number of stages instead of increasing k value by just 1 in each iteration. Therefore, one can skip the expensive calculations for those k values that are less likely to be optimum.

The proposed k -stage constraint strategy can be quite effective for the investment planning models because the trend in problems of this class is that their optimum solution involves investments in the earlier stages of the project. The reason behind this is the effect of economies of scale, as in general, it pays to make investments only once and earlier because of

the fixed cost charges. Second, if one expands the capacity, it is better to do it early as otherwise one will not take full advantage of the investment. This implies that the investment and operation decisions in the early stages of planning horizon are critical for these problems and require enforcing the NA constraints in these stages, while the ones for later stages can be ignored making the large-scale investment planning problems easier to solve. Specific examples for these problems are process network planning, or oil and gas fields infrastructure planning problems.

7.2 NAC Relaxation Strategy

The k -stage constraint strategy presented in the previous section involves the solution of the reduced model for the specified number of stages iteratively and has advantage for the investment planning problems where only first few stages involve uncertainty realization. This is due to the economies of scale in these problems as explained earlier. On the other hand, if there are endogenous uncertainties that are revealed later in the planning horizon, then the stage constraint approach can become expensive for finding the optimal solution due to the multiple solution of MILP problems, although strong lower bound to the problem can still be obtained.

Therefore, for the more general problems we propose a NAC relaxation strategy. This strategy is motivated by the fact that very few inequality NA constraints become active at the optimal solution of the problem (e.g. see Colvin and Maravelias, 2010). In this strategy (Figure 3), we divide the solution procedure in two phases, Phase I and Phase II. Phase I involves removing all inequality NA constraints from the reduced model (MSSP^R) and solving its LP relaxation (LP-MSSP^R). Then we check the feasibility of the NA constraints and add the violated NA constraints in the LP relaxation and solve iteratively until there is no violation of the NA constraints in the LP relaxation. In Phase II of the NAC relaxation strategy, the resulting model from Phase I with the added cuts is solved as an MILP problem to obtain a lower bound that is usually very tight. The upper bound is obtained by fixing the binary decisions in the reduced model (MSSP^R) using the solution of the lower bounding MILP problem such that NA constraints are not violated and solving the problem in fullspace. If the gap between lower and upper bounds is more than the specified tolerance, we check the feasibility of the NA constraints for the MILP solution in the current iteration and solve the new MILP problem with violated NA constraints that serve as added cuts in the next iteration. The procedure of solving lower and upper bounding problems in Phase II continues until the gap between upper and lower bound is within the specified optimality tolerance. Note that in comparison to the branch and cut solution

method by Colvin and Maravelias (2010), the proposed NAC relaxation strategy is much easier to implement using the available commercial solvers directly, although there might be some trade-offs between these solution strategies in terms of the solution times.

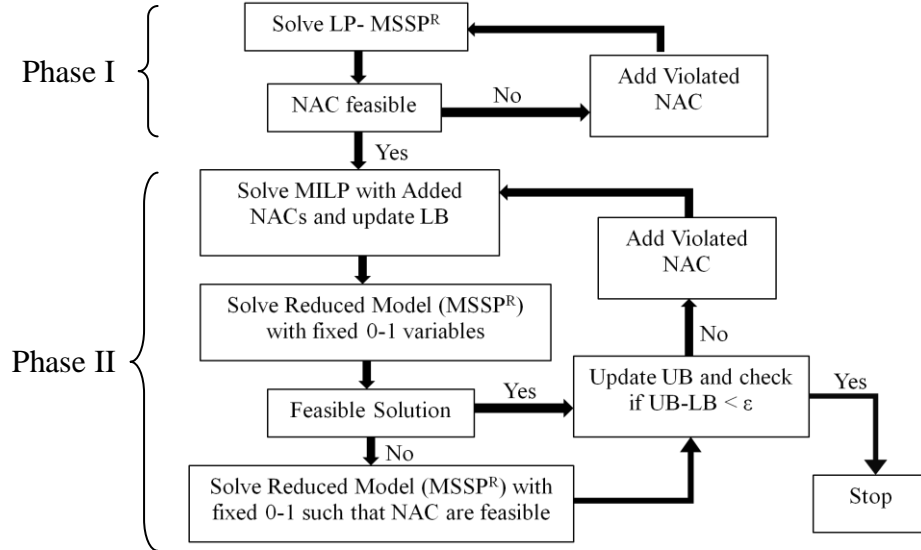


Figure 3: NAC Relaxation Strategy

It has been observed that very few inequality NA constraints (~6-7%) are added as cuts in the complete solution procedure and most of the violated NA constraints as cuts are added in Phase I itself which is very fast compared to Phase II. Although, the most expensive part of this procedure is the solution of the MILP problems during the Phase II iterations, it has been observed in most of the cases that only one or two iterations are required in Phase II to obtain a strong lower bound as well as the to generate the optimal solution from it. Moreover, due to the very small problem size compared to the reduced model, the solution of the MILP problems are significantly faster during these iterations.

7.3 Lagrangean Decomposition Algorithm

The solution strategies presented in the previous two sections basically require the solution of fullspace model and do not take the advantage of the decomposable structure of the model by scenarios. We should notice that the reduced model (MSSP^R) is composed of scenario subproblems connected through initial and conditional NA constraints. If these NA constraints are relaxed or dualized, then the problem decomposes by scenarios, and each sub-problem can be solved independently within an iterative scheme for the multipliers as described in Caroe and

Schultz (1999) and in Goel and Grossmann (2006). In this way, we can effectively decompose the large scale problems in this class.

In the Lagrangean Decomposition algorithm (Figure 4) the lower bound (LB) is obtained by solving the Lagrangean problem with fixed multipliers that is obtained from the reduced model (MSSP^R) by relaxing the conditional NA constraints and dualizing the first time period NA constraints as penalty terms in the objective. Each sub-problem in the following Lagrangean problem (LR^R-MSSP^R) corresponds to a scenario:

$$\begin{aligned}
(\mathbf{LR}^{\mathbf{R}}\text{-MSSP}^{\mathbf{R}}) \quad & \min \sum_{s \in S} p^s \sum_{t \in T} \left(c_t^{ws} w_t^s + c_t^{xs} x_t^s + c_t^{ys} y_t^s + \sum_{i \in I} c_{i,t}^{bs} b_{i,t}^s \right) \\
& + \sum_{(s,s') \in P_3} \sum_{i \in I} (\lambda_{b,i,1}^{s,s'} (b_{i,1}^s - b_{i,1}^{s'})) + \sum_{(s,s') \in P_3} (\lambda_{y,1}^{s,s'} (y_1^s - y_1^{s'})) \\
s.t. \quad & \sum_{\substack{\tau \in T, \\ \tau \leq t}} \left(A_{\tau,t}^{ws} w_{\tau}^s + A_{\tau,t}^{xs} x_{\tau}^s + A_{\tau,t}^{ys} y_{\tau}^s + \sum_{i \in I} A_{i,\tau,t}^{bs} b_{i,\tau}^s \right) \leq a_t^s \quad \forall s \in S, \forall t \in T \quad (11) \\
& w_t^s \in W_t^s, x_t^s \in X_t^s, y_t^s \in Y_t^s, b_{i,t}^s \in \{0,1\} \quad \forall s \in S, \forall t \in T, \forall i \in I \quad (2) \\
& Z_t^{s,s'} \in \{True, False\} \quad \forall s, s' \in S, \forall t \in T
\end{aligned}$$

The upper bound (UB) is generated by using a heuristic based on the solution of the Lagrangean problem. In this heuristic, we fix the decisions obtained from the above problem (LR^R-MSSP^R) in the reduced problem (MSSP^R) such that there is no violation of NA constraints and solve it to obtain the upper bound. The sub-gradient method by Fisher (1985) is used during each iteration to update the multipliers for the Lagrangean problem. The algorithm stops when either a maximum iteration limit is reached, or the difference between the bounds LB and UB is less than a pre-specified tolerance.

The major advantage with the above Lagrangean decomposition algorithm for endogenous uncertainty problems is that it provides good bounds on the optimal solution at the root node by taking advantage of the decomposable structure of the problem. Notice that in contrast to the method presented by Goel and Grossmann (2006), no branch and bound method is performed here with which the dual gap may not be closed for the problem. Therefore, if the gap between lower and upper bounds is large then in principle we would have to also incorporate a branch and bound procedure to reduce this gap. In our experience, however, we have observed for the problems in this class that the feasible solution obtained at the root node with this algorithm is also the optimal solution to the original problem with small optimality gap.

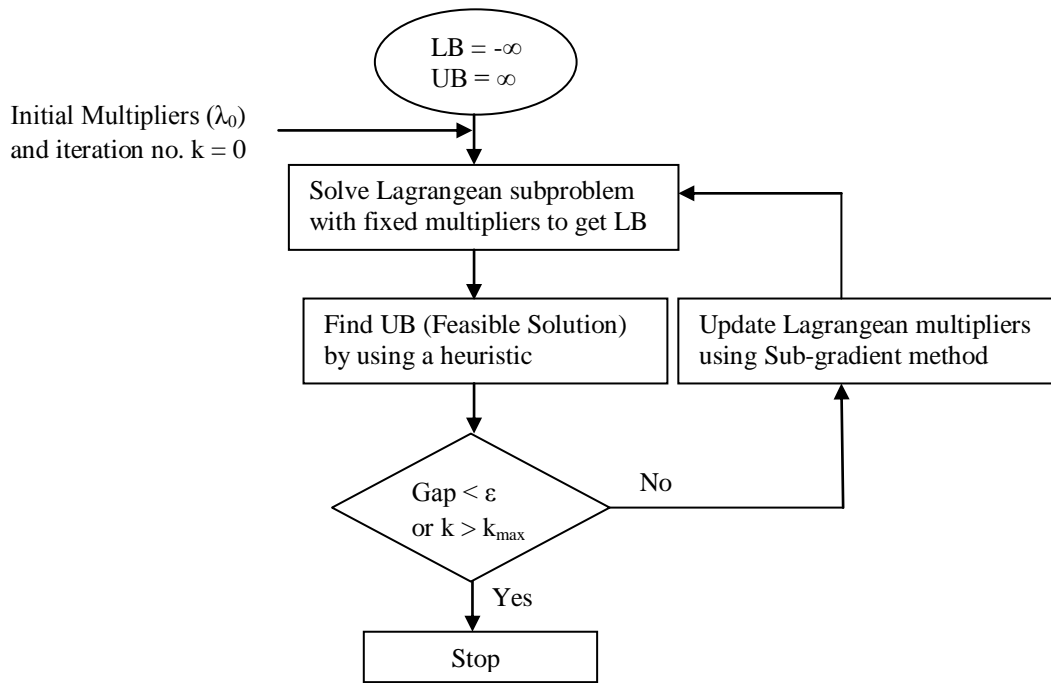


Figure 4: Lagrangean Decomposition algorithm

It should be noted that as opposed to the k -stage constraint method described earlier, in both the NAC relaxation strategy and Lagrangean decomposition algorithm, it is possible to assess the quality of the solution obtained (UB) with the lower bound at each iteration. On the other hand, in the k -stage constraint strategy we obtain the solution with optimal number of stages k .

8 Numerical Results

In this section we apply the proposed solution strategies to two process network examples and examine their performance compared to the original and reduced models.

8.1 Example 1

To illustrate the application of the various solution strategies for multistage stochastic programming with endogenous uncertainties, we consider the following problem from Goel and Grossmann (2006). Given is a process network (Figure 5) that is used to produce product A. Currently, the production of A takes place only in Process III with installed capacity of 3 tons/hour that consumes an intermediate product B that is purchased. If needed, the final product A can also be purchased so as to maintain its inventory. The demand for the final product, which is known, must be satisfied for all time periods over the given time horizon. Two new technologies (Process I and Process II) are considered for producing the intermediate B from two

different raw materials C and D. These new technologies have uncertainty in the yields. The yield of Process I and Process II can be (0.67,0.69,0.81,0.83,0.84) and (0.62,0.65,0.85,0.88,0.89), respectively, with equal probability of 0.2. These five realizations of yield for each of Process I and Process II give rise to a total of 25 scenarios.

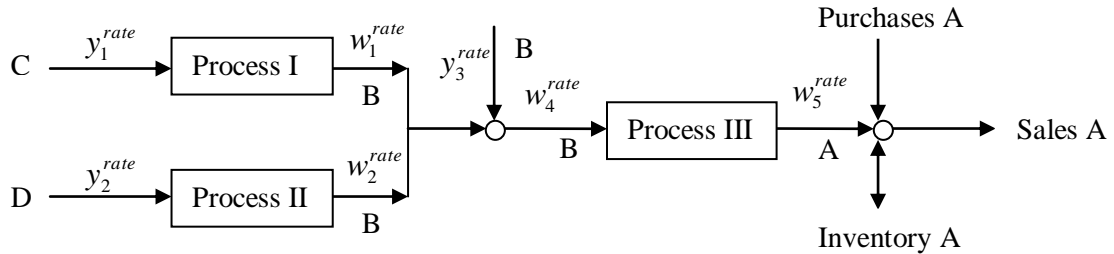


Figure 5: Process Network Example 1

The problem consists of finding the optimum expansion and operation decisions for this process network for a 10 year planning horizon to minimize the total expected cost of the project. Applying the original model (MSSP⁰) and solving it with XPRESS 20.00, we obtain the results shown in Figures 6(a)-(e). The total expected cost is \$369,124 and the solution suggests to install Process II with a capacity of 1 tons/hr and expand the existing Process III from a capacity of 3 tons/hr to 6.914 tons/hr in the first year. If the yield of Process II turns out to be low, i.e. 0.65 (Figure 6-a) or 0.62 (Figure 6-b), then in the second year it is not expanded and the new Process I is installed. On the other hand, if yield of Process II turns out to be high, i.e. 0.89, (Figure 6-c), 0.88 (Fi,g. 6-d) or 0.85 (Figure 6-e), then Process II is expanded in the second year to slightly different capacities close to 8 tons/hr in each of these three cases and there is no installation of Process I. There are no further installations or expansions of any of the processes.

It is interesting to note that the solution of the two-stage stochastic model of this example that considers no expansions, i.e. no recourse actions for the investment decisions of the processes, yields an expected cost of \$379,706 or about 3% higher than the multistage model. In this case the solution suggests to install Process I and Process II with capacities of 4.246 tons/hr, and 4.541 tons/hr respectively, and expand Process III to a capacity of 7.384 tons/hr in the first year. The savings in the expected cost using the multistage stochastic model are due to the fact that multistage stochastic solution takes advantage of favorable scenarios corresponding to the high yields of Process II, while minimizing the losses due to the low yield of Process II by taking appropriate recourse action in the future on the basis of the outcome of the yield of Process II. It can be seen from the Figures 6 (c)-(e), that there is no investment made in Process-I for the

scenarios corresponding to the high yields of Process II and from Figures 6(a)-(b), that there is installation of Process I for the scenarios corresponding to low yields of Process II.

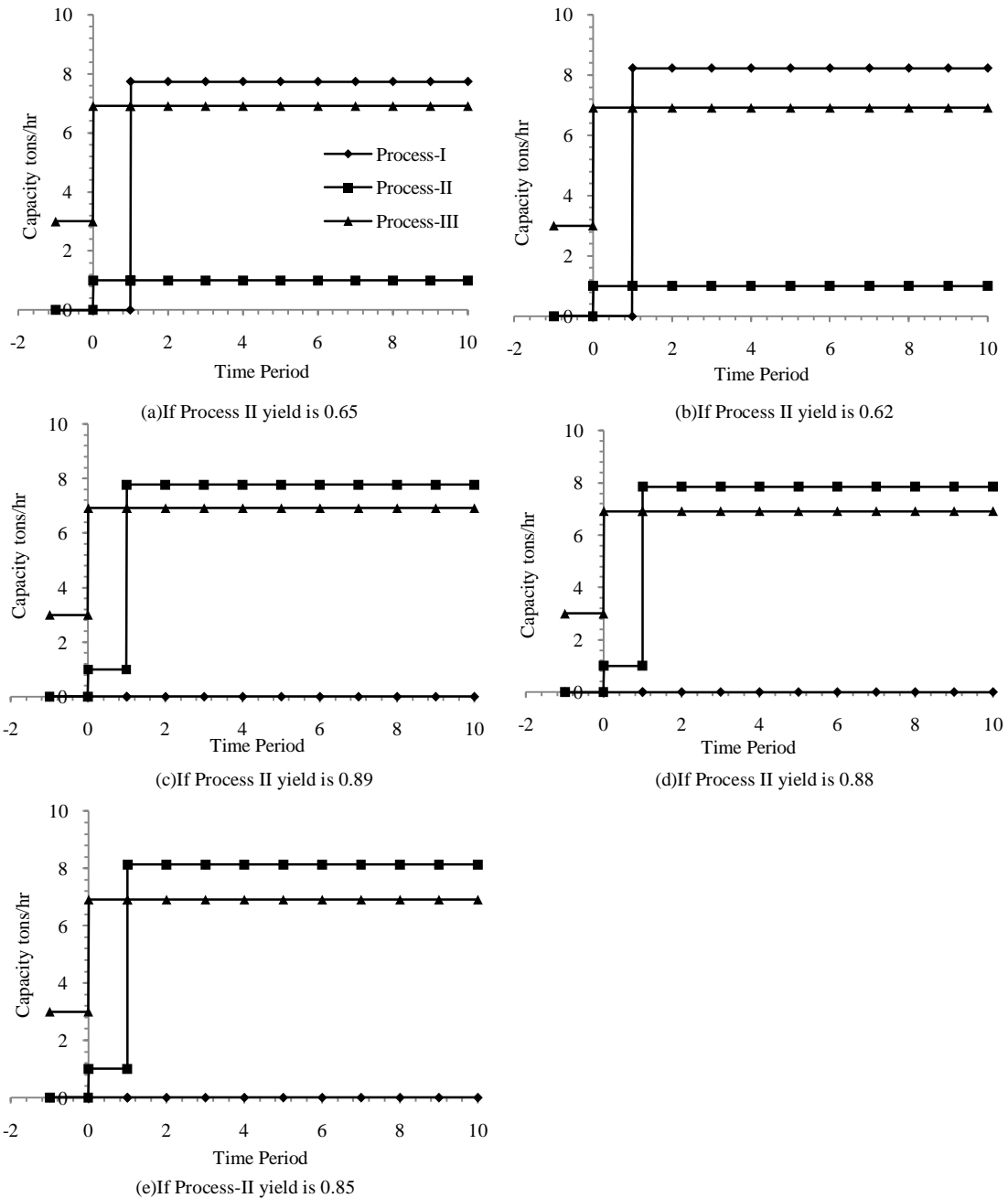


Figure 6: Installation Schedule for the Process Network Example 1

If we reformulate the original (MSSP⁰) and reduced (MSSP^R) models into MILP problems (see Goel and Grossmann, 2006) for this example, the comparison of problem sizes and solution times between these models using XPRESS 20.00 solver is given in Table 3. It can

be seen that problem size has reduced approximately 90% using Properties 1-3. Therefore, the advantage of including the new Property 3 with the earlier Properties 1 and 2 is very significant for this problem.

Table 3: Comparison of the various solution strategies for Example 1

Problem Type	(s, s')	Expected Cost (\$10 ³)	Number of Constraints	Continuous Variables	Binary Variables	Solution Time(s)
Original Model (MSSP ⁰)	600	369.12	192,376	11,026	750	243.33
Reduced Model: Property 1	300	369.12	98,576	8,026	750	224.79
Reduced Model: Properties 1-2	100	369.12	32,376	6,026	750	56.76
Reduced Model (MSSP ^R): Properties 1-3	40	369.12	15,816	5,426	750	35.94
<i>k</i> -stage constraint Model for <i>k</i> =2	40	369.12	7,096	5,106	750	8.36
NAC Relaxation Strategy	40	369.12	8,187*	5,426*	750*	12.00

*Size of the last MILP with NA constraints in Phase II.

** Solved using XPRESS 20.00 solver in GAMS 23.0 on an Intel Pentium-IV machine with 3 GB of RAM.

The comparison of the *k*-stage constraint strategy with the original (MSSP⁰) and the reduced (MSSP^R) models for this 3 process network is also given in Table 3 where it can be seen that the global optimum is obtained using the *k*-stage constraint strategy and the solution time is greatly decreased to only 8.4s. We should note that the problem was solved with 2-stages initially and was stopped after the first iteration itself because there was no installation in time periods after *k*=2, and therefore Proposition 2 is satisfied.

When the NAC relaxation strategy is applied to this problem, it provides the optimal solution significantly faster compared to the fullspace model as seen in Table 3, and its performance is slightly slower than the *k*-stage constraint strategy. The problem size of the MILP in the last iteration with this strategy after adding the violated NA constraints is also comparable to the size of the *k*-stage model.

The Lagrangean decomposition algorithm was also used for solving the process network Example 1 using the reduced model. The results in Table 4 show that with the Lagrangean decomposition algorithm the problem can be solved within 1% of optimality at the root node in just 27 s compared to 243 s in the case of the original model. Note that the global optimum is also obtained in this case. To further reduce the gap one may have to incorporate a branch and bound method.

Table 4: Iterations during Lagrangean Decomposition

Sub-gradient Iteration No.	Lower Bound (\$10 ³)	Upper Bound (\$10 ³)	% Gap
1*	360.408	369.124	2.361
2*	362.594	369.124	1.769
3*	363.795	369.124	1.444
4	363.795	369.124	1.444
5*	364.244	369.124	1.322
6	364.789	369.124	1.174
7*	364.816	369.124	1.167
8	364.883	369.124	1.149
9	364.883	369.124	1.149
10*	365.374	369.124	1.016
27	366.135	369.124	0.810
Time(s)	21.95	5.18	0.810

*problem solved for upper bound generation

Furthermore, to investigate the impact of the model reduction using Properties 1-3, we also consider other cases for this example where the number of realization of uncertain yields are changed for Process I and Process II from 2 to 10, and the results are shown in Table 5. It is clear that the problem size is reduced significantly and hence the solution time for all the cases. Also, note that we can solve all the problems with the reduced model, while the larger ones cannot be solved with the original model. The main reason is the much smaller size of the reduced model as can be seen in Figure 7.

Table 5: Comparison of the original and reduced models for Example 1 considering different scenarios

Number of Scenarios	Solution Time(s) Original Model*	Solution Time(s) Reduced Model*	% Optimality Gap
4	1.30	0.96	0
9	19.38	4.98	0
16	133.09	14.71	0
25	243.33	35.94	0
36	731.37	42.26	< 0.5%
64	2516.709	102.04	< 0.5%
81	NA	105.03	< 0.5%
100	NA	120.19	< 0.5%

*Problems are solved in fullspace.

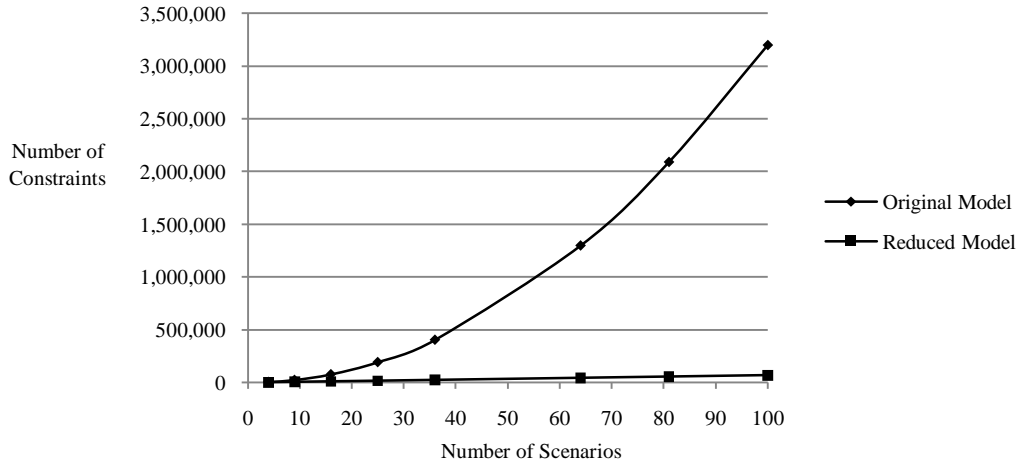


Figure 7: Comparison of constraints in Original and Reduced Models for Example 1 considering different scenarios

As discussed earlier, the number of active NA constraints at the optimal solution of these problems is very small. It can be observed from Figure 8 that very few (~6-7%) of the inequality constraints are added as cuts during the NAC relaxation strategy for all scenario instances of Example 1. Also, the computational advantage of this strategy can be seen in Table 6. It should be noted that very few Phase II iterations are needed to obtain the optimal solution.

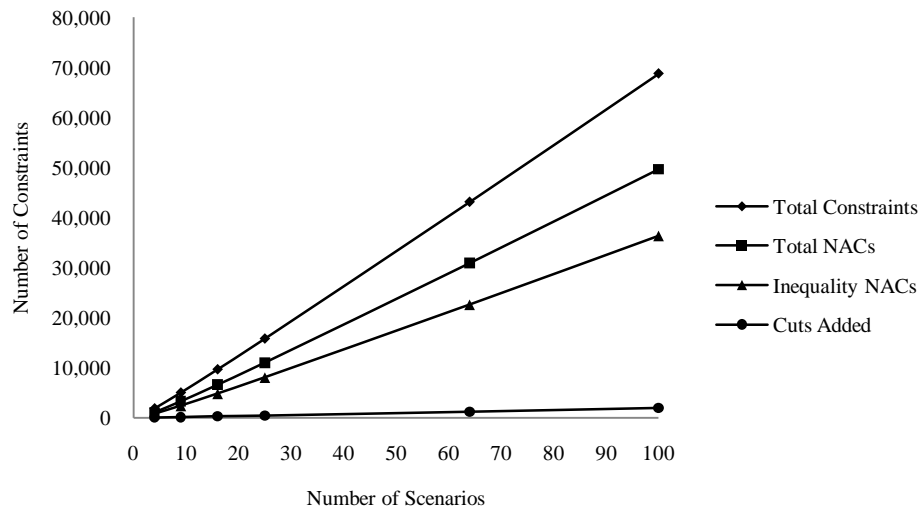


Figure 8: Cuts Added vs. Total Constraints in the Reduced Model for NAC Relaxation Strategy

Table 6: Reduced Model vs. NAC-Relaxation Strategy for various scenario instances

Number of Scenarios	Optimal Solution ³ (\$10 ³)	Gap %	Phase I Iterations	Phase II Iterations	Solution Time (s) NAC Relaxation Strategy	Solution Time(s) Reduced Model
4	379.072	0.000	3	1	1.136	0.96
9	390.944	0.012	3	1	3.701	4.98
16	377.364	0.002	3	1	10.837	14.71
25	369.124	0.002	3	1	12.005	35.94
64	376.824	0.000	5	2	51.577	102.04
100	376.747	0.003	3	1	76.537	120.19

8.2 Example 2

To illustrate the solution of a larger instance, we consider a 5 process network (Figure 9) having 4 uncertain parameters, i.e. yield of Process I, Process II, Process IV and Process V. Notice that here we consider 2 new additional processes compared to the previous example in which Process IV converts E into B and Process V that converts B into final product A. Each of the uncertain yields has 3 realizations and gives rise to a total of 81 scenarios with equal probabilities.

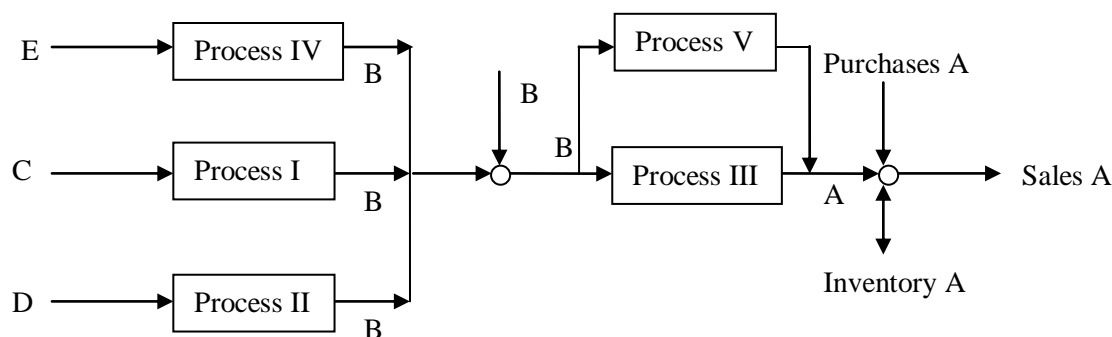


Figure 9: Process Network Example 2

The problem consists of finding the expansion and operation decisions for this process network over a 10 year planning horizon to minimize the total expected cost of the project. The optimum installation schedule of the processes for this problem can be seen in Figure 10. Only one node in time period 1 in Figure 10 corresponds to the initial state of the system when there is no realization of any of the uncertain yields. The uncertain Process II, Process IV and Process V are installed in the first year with small capacities in all the scenarios and due to the 3 possible realizations of the yield of each of these 3 processes, there are total 27 nodes at time period 2 in the scenario tree (Figure 10) that correspond to the 27 possible states of the system at the beginning of the second year. On the basis of these yield realizations, the recourse actions

involve installation of the new Process I for low yield scenarios and expansion of the already installed processes for high yield scenarios. Note that in Figure 10, the number of nodes (states) in time period 3 is greater than the ones in period 2 due to the installation of Process I in some of the states in the second year and its corresponding 3 possible yield realizations for each of these new installations. From period 3 to end of the planning horizon there is no further realization of uncertainty in any of the scenarios and no new branches appeared as can be seen from Figure 10.

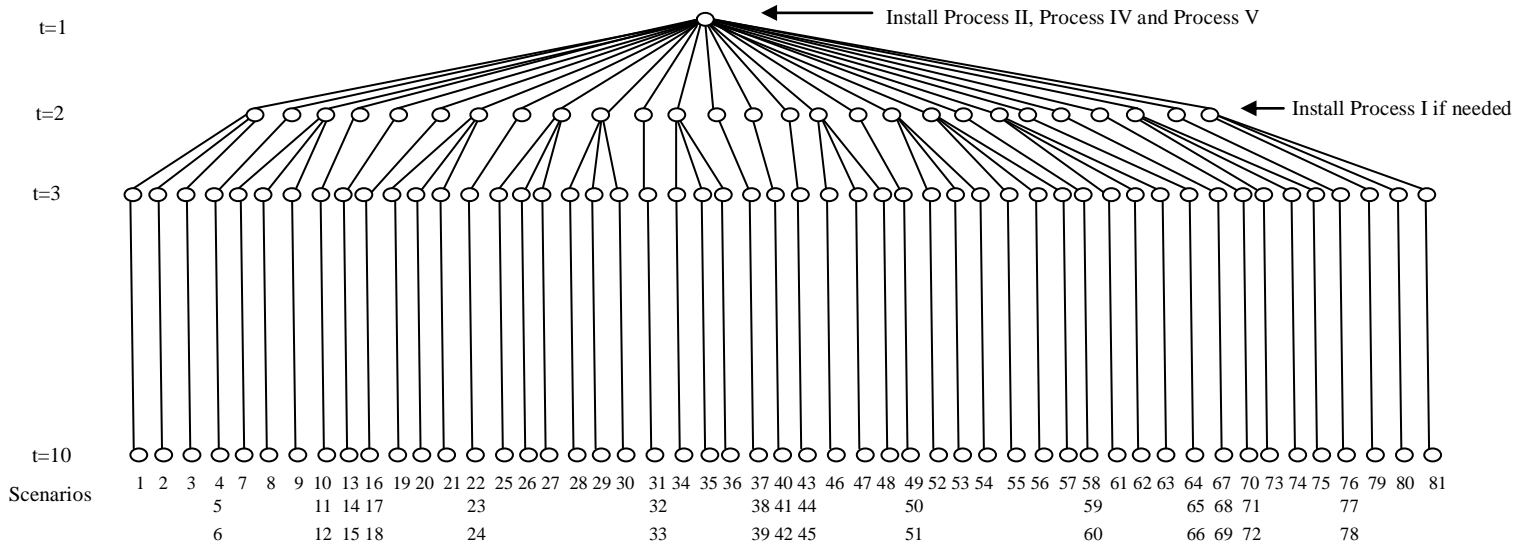


Figure 10: Optimal Solution for Example 2

The results for this problem are compared in Table 7 for the original (MSSP⁰), reduced (MSSP^R), k -stage constraint (MSSP^{sc}) models and NAC relaxation strategy. The problem was solved within 0.5% optimality tolerance in all the cases which gives slightly different optimal values. It can be seen that the problem cannot be solved in the fullspace for the original model and even after using Property 1, while using the reduced model with Properties 2 and 3, we can solve it. The solution time for only considering Properties 1-2 is 1.5 times more than the solution time from considering Properties 1-3, which is expected due to a factor of around the same order in the number of scenario pairs included in these models.

The k -stage constraint model was initially solved for two stages ($k = 2$) and it gives the optimal solution to the problem as there was no realization of any uncertain parameter after $k=2$. Because of the inherent property of these problems, the proposed k -stage constraint model does not need many iterations and performs better than the reduced model. On the other hand, the NAC relaxation strategy works well in all the cases because of its generality. As it can be seen in

Table 7, the optimal solution obtained from the NAC relaxation strategy has a slightly lower cost than the other strategies, and it is also significantly faster than the reduced model and comparable to the k -stage model.

Table 7: Comparison of the various solution strategies for Example 2

Problem Type	(s,s')	Expected Cost (\$10 ³)	Number of Constraints	Continuous Variables	Binary Variables	Solution Time(s)
Original Model (MSSP ^d)	6480	-	3,158,272	90,802	4,050	NA
Reduced Model: Property 1	3240	-	1,591,732	58,402	4,050	NA
Reduced Model: Properties 1-2	324	369.590	151,552	29,242	4,050	1627.51
Reduced Model (MSSP ^R): Properties 1-3	216	368.972	109,432	28,162	4,050	1160.34
k -Stage Constraint Model for $k=2$	216	368.916	44,200	26,434	4,050	371.53
NAC Relaxation Strategy	216	368.650	45,797*	28,162*	4,050*	250.64

*Size of the last MILP with NA constraints in Phase II.

The Lagrangean decomposition algorithm was also applied to this 5 process network problem using the reduced model. The results in Table 8 show that using Lagrangean decomposition algorithm with the reduced model, the problem can be solved within about 3.5% of optimality gap at the root node after 30 iterations. The solution obtained (UB) at the root node has a higher cost than the solution obtained from the NAC relaxation strategy (\$371,579 vs. \$368, 650). On the other hand, it is faster than the NAC relaxation strategy (181s vs. 251s).

Table 8: Iterations during Lagrangean Decomposition algorithm

Sub-gradient Iteration No.	Lower Bound (\$10 ³)	Upper Bound (\$10 ³)	% Gap
1*	351.577	371.579	5.383
2*	352.517	371.579	5.130
3*	354.426	371.579	4.616
4*	354.426	371.579	4.616
5*	354.869	371.579	4.497
6*	354.869	371.579	4.497
7*	354.929	371.579	4.481
8*	354.929	371.579	4.481
9	355.235	371.579	4.399
10	355.235	371.579	4.399
30	358.361	371.579	3.557
Time(s)	167.19	13.63	3.557

*problem solved for upper bound generation

The Lagrangean decomposition strategy has the advantage that if the problem size is too large to be generated for all the scenarios at once, the model can be decomposed by scenarios. The k -stage constraint and NAC relaxation strategies will not work in this case as they need to be solved for all scenarios at once. It is only in smaller to moderate size problems that the k -stage constraint strategy and the NAC relaxation strategy may perform better than Lagrangean decomposition strategy because of the tight lower bounds and corresponding better solutions obtained in these cases. These trends can be clearly seen from the two examples considered.

It is also interesting to note that the two-stage stochastic model corresponding to this example gives about 5% higher total expected cost (\$387,421 vs. \$368,650) and suggests to invest in all the processes in period 1. Similar to the Example 1, in the two-stage case the higher cost occurs due to the absence of appropriate recourse for the investment decisions in the model. Furthermore, the larger savings compared to the previous example indicate the advantage of using the multistage stochastic model. Also, note that the total expected cost is about 3-6 % higher for the expected value problem (EVP) in comparison to the multistage stochastic programming model for all the cases considered.

The numerical results presented in this section are very encouraging to solve multistage stochastic programming problems with endogenous uncertainty using the proposed solution strategies in reasonable computational time. Although there are several trade-offs involved in using a particular solution strategy for a particular class of the problems under uncertainty, the proposed solution strategies are fairly general and can be applied to many problems classes, specifically to all the problems that involve endogenous uncertain parameters.

9 Conclusions

In this paper, we have proposed several solution strategies for Multistage Stochastic Programming problems with endogenous uncertainty. We have identified a new Property 3 for the models in this class that together with two properties previously presented by Goel and Grossmann (2006), significantly reduce the problem size and the solution time. To solve the large instance of these problems, we have proposed a k -stage constraint strategy that yields the global optimum in particular cases and is useful for problems where endogenous uncertainty is revealed during the first few time periods of the planning horizon. To solve the more general problems of large size, we also proposed a NAC relaxation strategy based on relaxing the NA

constraints and adding them if they are violated. Finally, we described a Lagrangean Decomposition algorithm that can predict the rigorous lower bounds for the solution obtained. The proposed solution strategies have been successfully applied to two process network problems. Moreover, these strategies are applicable to a wide range of problems having endogenous uncertainty in some of the parameters.

10 Acknowledgements

The authors acknowledge to ExxonMobil Upstream Research Company for the financial support of this work.

References

- Ahmed, S., 2000. Strategic planning under uncertainty: Stochastic integer programming approaches. Ph.D. thesis, University of Illinois at Urbana-Champaign.
- Birge, J. R., Louveaux, F., 1997. Introduction to stochastic programming. New York, NY: Springer.
- Boland, N., Dumitrescu, I., Froyland, G., 2008. A multistage stochastic programming approach to open pit mine production scheduling with uncertain geology.
<http://www.optimizationonline.org/DBHTML/2008/10/2123.html>.
- Caroe, C.C., Schultz, R., 1999: Dual decomposition in stochastic integer programming. Oper. Res. Lett. 24, 37–45.
- Clay, R. L., Grossmann, I. E., 1997. A disaggregation algorithm for the optimization of stochastic planning models. Computers and Chemical Engineering 21 (7),751-774.
- Colvin, M., Maravelias, C. T., 2008. A stochastic programming approach for clinical trial planning in new drug development. Computers and Chemical Engineering 32 (2008) 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 (2010) 205–215.

- Fisher, M. L., 1985. An Applications Oriented Guide to Lagrangian Relaxation. *Interfaces* 15, 10-21.
- 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., Mulkey, 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.
- Held, H., Woodruff, D. L., 2005. Heuristics for multi-stage interdiction of stochastic networks. *Journal of Heuristics* 11 (5-6), 483-500.
- Ierapetritou, M. G., Pistikopoulos, E. N., 1994. A novel optimization approach of stochastic planning models. *Industrial and Engineering Chemistry Research* 33, 1930-1942.
- 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.
- Jonsbraten, T., 1998. Optimization models for petroleum field exploitation. PhD thesis, Norwegian School of Economics and Business Administration.
- Jonsbraten, 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.
- Pflug, G. Ch., 1990. Online optimization of simulated markovian processes. *Mathematics of Operations Research Res.* 15, No.3, 381-395.
- Ruszczynski, A., 1997. Decomposition methods in stochastic programming. *Math. Programming (Ser. B)* 79, 333-353.
- Sahinidis, N.V., 2004. Optimization under uncertainty: State-of-the-art and opportunities. *Comput. Chem. Eng.* 28 (6–7), 971–983.

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

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

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. *Industrial and Engineering Chemistry Research* 48 (6), 3078-3097.

Viswanath, K., Peeta, S., Salman, S. F., 2004. Investing in the links of a stochastic network to minimize expected shortest path length. Tech. rep., Purdue University.