A Multistage Stochastic Programming Approach with Strategies for Uncertainty Reduction in the Synthesis of Process Networks with Uncertain Yields

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

This paper is dedicated to Professor Reklaitis
for his leadership in the area of Process Systems Engineering.

ABSTRACT

In this paper we consider the synthesis of process networks with time-varying uncertain yields in which investment in pilot plants can be considered to reduce uncertainty of the yields. We formulate this problem as a multistage stochastic program with decision dependent elements where investment strategies are considered to reduce uncertainty, and time-varying distributions are used to describe uncertainty. We propose a new mixed-integer/disjunctive programming model which is reformulated as a mixed-integer linear program. Since the model can only be solved through an LP-based branch and bound for smaller instances, we propose a duality-based branch and bound algorithm for solving larger problems. A numerical example is presented to illustrate the application of the proposed method.

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

1. INTRODUCTION

Synthesis problems require making decisions in the presence of uncertainty. Recently, modeling and solution of this class of problems with stochastic programming is receiving more attention because of the increasing importance of anticipating the effects of uncertainty (for a recent review of stochastic programming, see Birge & Louveaux, 1997). There is a large amount of literature on uncertainty because many disciplines, ranging from economics to engineering need to account for the effect of uncertainties in their respective fields (see Sahinidis, 2004).

According to Jonsbraten (1998), uncertainty in planning problems can be divided into two classes: exogenous uncertainty and endogenous uncertainty. Problems where stochastic processes are independent of decisions are said to have exogenous uncertainty (e.g. demands), whereas problems where stochastic processes are affected by decisions are said to possess endogenous uncertainty (e.g. yields). In this paper we address a synthesis problem that involves endogenous uncertainty.

Literature on the class of problems that deal with endogenous uncertainty is limited. The only papers that we are aware of are by Pflug (1990); Jonsbraten, Wets & Woodruff (1998); Jonsbraten (1998); Ahmed (2000); Held & Woodruff (2003); Vishwanath, Peeta & Salman (2004); Goel & Grossmann (2004, 2005). Pflug (1990) addresses the problems in the context of discrete event dynamic systems where the underlying stochastic process depends on the optimization decisions. Jonsbraten et al. (1998) address endogenous uncertainty where project decisions reduce the uncertainty. These authors have proposed an implicit enumeration algorithm where decisions that affect the uncertain parameter values are made at the first stage. Ahmed (2000) presents examples on network design, server selection and facility location problem where decision dependent uncertainties take place. The author shows that these programs can be reformulated as MILP problems and solved by LP-based branch & bound algorithms. Held & Woodruff (2003) have worked on another instance where the problem has endogenous uncertainty in the structure of network. In each stage of the problem, an operator tries to find the shortest path from a source to destination after the interdictor interdicts some of the nodes in the network. The aim is to maximize the probability of stopping the flow goods or information in the network. Vishwanath, Peeta & Salman (2004) addressed a network problem having endogenous uncertainty in survival distributions. The problem is a two stage stochastic program where first period investment decisions are made for changing the survival probability distribution of arcs after a disaster. The aim is to find the investments that will minimize the expected shortest path from source to destination after a disaster. Goel & Grossmann (2004) dealt with the gas field development problem under uncertainty in size and quality of reserves where decisions on the timing of field drilling are assumed to yield an immediate resolution of the uncertainty. In their solution strategy, the authors used a relaxed problem to find upper bounds and solve a multistage stochastic programs for fixed scenario tree for finding lower bounds. Goel & Grossmann (2005) later proposed a branch and bound algorithm where lower bounds are generated by Lagrangean duality. They also applied this general formulation to two problems: capacity expansion of process networks and sizing of components for manufacturing.

Another aspect that will be addressed in this paper is gradual reduction of uncertainty over time. The literature about this subject in the context of synthesis and planning problems is also very limited. The only literature that we know in the context of planning problems is Stensland & Tjøstheim (1991), Jonsbraten (1998) and Dias (2002). Stensland & Tjøstheim (1991) have worked on a discrete time problem for finding optimal decisions with uncertainty reduction over time and they applied their approach to oil production. These authors expressed the uncertainty in terms of a number of production scenarios. Their main contribution was combining production scenarios and uncertainty reduction effectively for making optimal decisions. Jonsbraten (1998) presents gradual uncertainty reduction where all uncertainty is assumed to resolve at the end of the project horizon. They use a decision tree approach for modeling the problem where Bayesian statistics are applied to find the probabilities of branches in the decision tree that are decision dependent. The author also proposes an algorithm that relies on the prediction of upper and lower bounds. Dias (2002) presents four propositions to characterize technical uncertainty and the concept of revelation towards the true value of the variable. These four propositions, based on the theory of conditional expectations, are employed to model technical uncertainty.

The outline of this paper is as follows. In section 2 we present the general problem statement for the class of problems under consideration and then in section 3, we discuss the representation of scenarios. In section 4, a motivating example problem is given. In section 5 and 6, generalized disjunctive model and the proposed solution approach are explained. Section 7 discusses the results found by the proposed solution approach and compares the results with other solutions.

2. PROBLEM STATEMENT

In this paper, the problem that we consider is the multi-period optimization for process networks under gradual uncertainty resolution in the process yields with possible investments in pilot plants for reducing uncertainties.

Given a process network with availabilities and demands of raw materials, intermediates and final products over T time periods, the problem is to determine in each time period t whether the capacity of specific processes should be expanded or not, whether specific processes should be operated or not, and whether pilot plants for specific processes should be installed or not. In addition, other decisions include selecting the actual expansion capacities of the processes, the input flow rates into the network, and the amount of purchase and sales of final product. The objective is to make these decisions so as to maximize the expected net present value.

The major uncertainties considered in the above problem are the yields of the processes, which as discussed in the introduction are endogenous parameters. We assume that the time-varying uncertainty reduction is contingent on the operation and pilot plant installation decisions over the time horizon. Uncertainty is reduced in two different ways (see Figure 1). One way is installing the actual plant directly and observing how the yields reveal over time periods (Figure 1a). Another way is investing in a pilot plant and conduct experiments (e.g. for one time period) in order to gain more accurate knowledge of the uncertainty (Figure 1b). One can then invest in a plant, starting at a reduced level of uncertainty. In the second option, we shield the process from a larger uncertainty but delay large scale production for one time period.

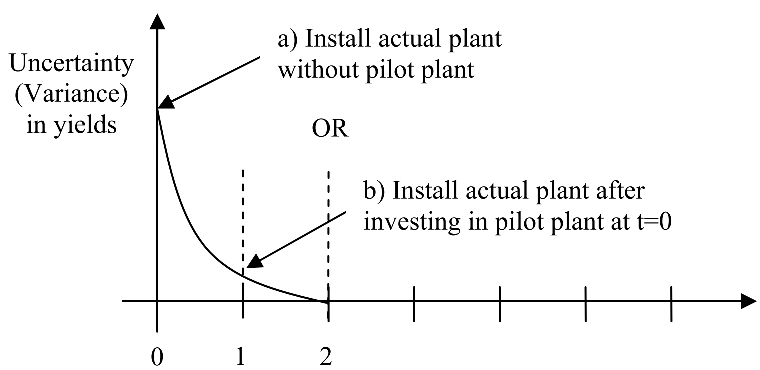

Figure 1: Options for uncertainty reduction.

Note that in Figure 1, we assume for simplicity that the reduction in uncertainty from installing and operating the plant for one time period is identical to the reduction by conducting experiments with the pilot plant for one time period. Also, for the sake of simplicity, we will assume that the decrease in time varying uncertainty will be completed on finite number of time periods. This simplification limits the number of scenarios needed to represent the gradual uncertainty as will be explained in the representation of scenarios section. Also, we will assume that the duration for pilot plant is equivalent to one period of operation and that a pilot plant can operate only once during the entire project horizon.

A general model for synthesis of process networks under uncertainty is presented next and applied to a specific example. As will be shown, the variables are discrete (binary decision variables) and continuous and the constraints involve linear equations and disjunctions. Therefore the model falls under the general class of linear mixed-integer/disjunctive programming problems. Before presenting the model, which is motivated by the work of Goel and Grossmann (2005), we describe the way in which scenarios will be represented.

3. REPRESENTATION OF SCENARIOS

In this paper, the time horizon is discretized into time periods and the probability distributions of uncertain parameters are discrete because these specifications allow us to represent the stochastic process by scenario trees. Figure 2 is a standard representation of a scenario tree having one uncertain parameter 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, and the four scenarios have equal probabilities as all the transition probabilities are identical.

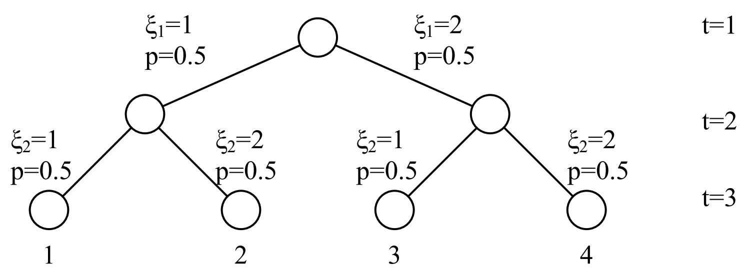

Figure 2: Scenario tree with uncertain parameters $\xi 1$and $\xi 2$.

In a standard scenario tree, each node for time period t represents a possible state for 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 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. The set of time periods, which has the same amount of information, define a stage. Problems that have more than two stages are called multi-stage stochastic programs.

Figure 3 is an alternative representation of the scenario tree in Figure 2, proposed by Ruszczynski, 1997. In this representation, each scenario is represented by a set of unique nodes. The horizontal lines connecting nodes in time period t mean that these nodes are indistinguishable and they have the same amount of information in that time period. The horizontal lines reduce the tree in Figure 3 to the one in Figure 2. For modeling the problem, the scenario trees will be considered according to this representation given by Ruszczynski (1997) (see Figure 3).

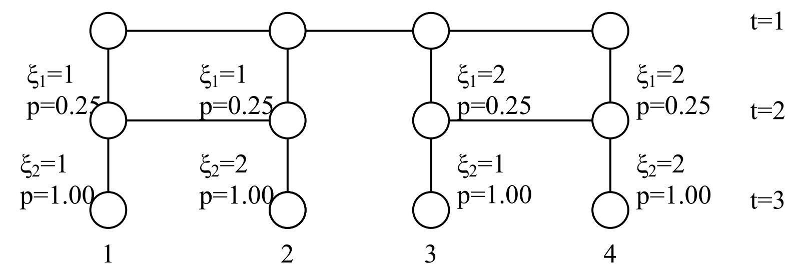

Figure 3: Alternative scenario tree with uncertain parameters ξ1and ξ2.

As mentioned before, we assume that the uncertainties in yields are described by time-varying discrete distributions and the discrete distributions of yields give rise to scenario trees. Representation of time-varying distributions using scenario trees is a crucial point. For simplicity, we assume that each uncertain yield is resolved over two time periods in three steps. Step 1 is the one where no investment is made for reducing uncertainty. Step 2 is the step where uncertainty is resolved partially. Step 3 is the last step where uncertainty is fully resolved. In terms of time periods, Step 1 is the beginning of time period t, Step 2 is the end of time period t and the beginning of time period t+1 and Step 3 is the end of time period t+1 (Figure 4).

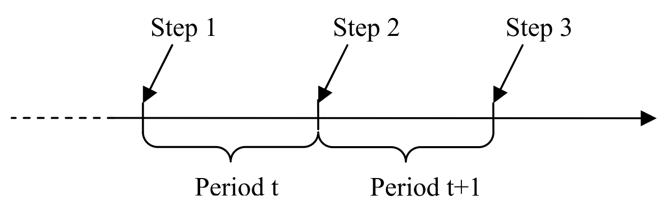

Figure 4: Relation between steps and periods.

It should be noted that there is not a unique way of representing time-varying uncertainties and following are three possible representations. A possible scenario tree representation for gradual uncertainty resolution in three steps is given in Figure 5. It is assumed that every possible yield (60, 70, 80 and 90%) can be realized in step 1 and step 2 and the yield can switch from one to another until step 3 where all uncertainty resolves. The disadvantage of this representation is that the number of scenarios increases rapidly which eventually increases the model size. Therefore, some simplifications can be done on this representation based on the relative transition probabilities between states. For instance, if we assume that the transition probability from 90% in step 2 to 60 or 70% is smaller compared to other transition probabilities, we can simply disregard that transition so we end up with only 2 branches from 90% in step 2 to 80 and 90% in step 3. The following possible representations are actually based on similar arguments to simplify this scenario tree.

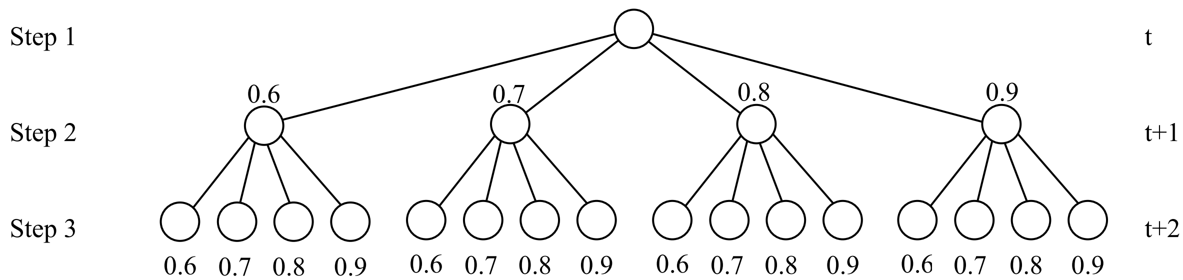

Figure 5: Example scenario tree representation (1).

Another possible scenario tree representation is given in Figure 6. In this representation, in step 2, where uncertainty is partially resolved, all possible yields appear (60, 70, 80 and 90%). These yields become certain in step 3 when the uncertainty is totally resolved. For simplicity, in step 2, the distribution is divided into two parts and a realized yield will be on one of these parts. When all the uncertainty is revealed, the exact value will be one of the possible values on the same part of the value realized in step 2. For example, if the realized yield is 70% in the step 2, it can be 60% or 70% in step 3 but cannot be 80% or 90% as those values lie on the other part of the distribution.

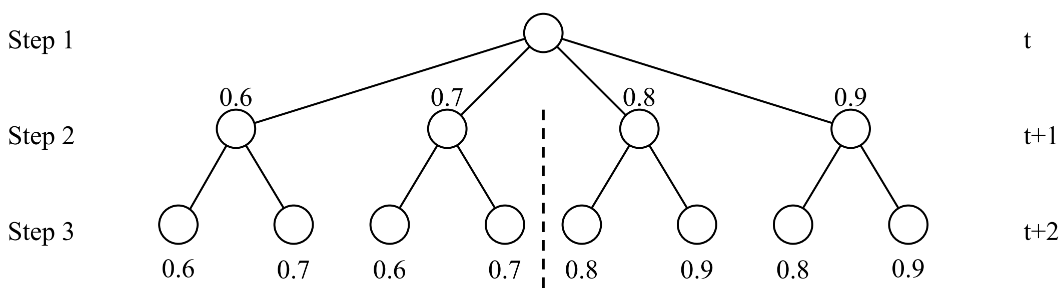

Figure 6: Example scenario tree representation (2).

The third possible scenario tree representation is shown in Figure 7, where in the partially revealed uncertainty at step 2, only realizable yields are the highest and the lowest of all possible values (i.e. 60% and 90%). When the uncertainty is totally revealed, all possible yields are realized. In the figure there are two possible realizations for yields in step 2 and four possible yields in step 3. It is clear that the tree structure in Figure 7 is a special case of the tree structure in Figure 6 where intermediate yields cannot be realized in step 2. The disadvantage of the tree representation in Figure 5 and 6 is that the number of scenarios in the resulting model is larger than the one in Figure 7. Therefore, for convenience we adapt in this work this third scenario tree representation, although the proposed model can be easily extended to handle the scenario trees of Figures 5 and 6.

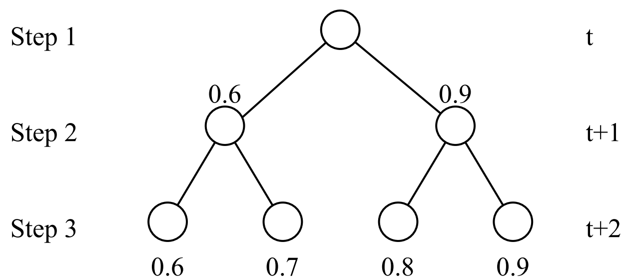

Figure 7: Example scenario tree representation (3).

Also, we should note that if there are continuous distributions describing uncertain parameters, they can be represented in discrete form and used during generation of scenario trees. One of the possible ways to transform these continuous distributions into discrete ones is explained in Luceno (1999). In this case, we do not have to use a fixed set of possible yield values, but we can use different values based on the discretization of the continuous distribution at each step.

4. MOTIVATING EXAMPLE

As an illustration, Figure 8 shows an example of a process network that can be used to produce a given product. Currently, the production of A takes place only in process 3, which consumes an intermediate product B from the market for production. If needed, the final product A can also be purchased from the market and its inventory can be maintained. The demand for the final product, which is assumed to be known, must be satisfied for all periods over the given time horizon. Two new technologies (process 1 and process 2) are available to produce the intermediate product from two different raw materials C and D. These new technologies have uncertainty in the yields which is reduced with operation time, or with investments in pilot plants.

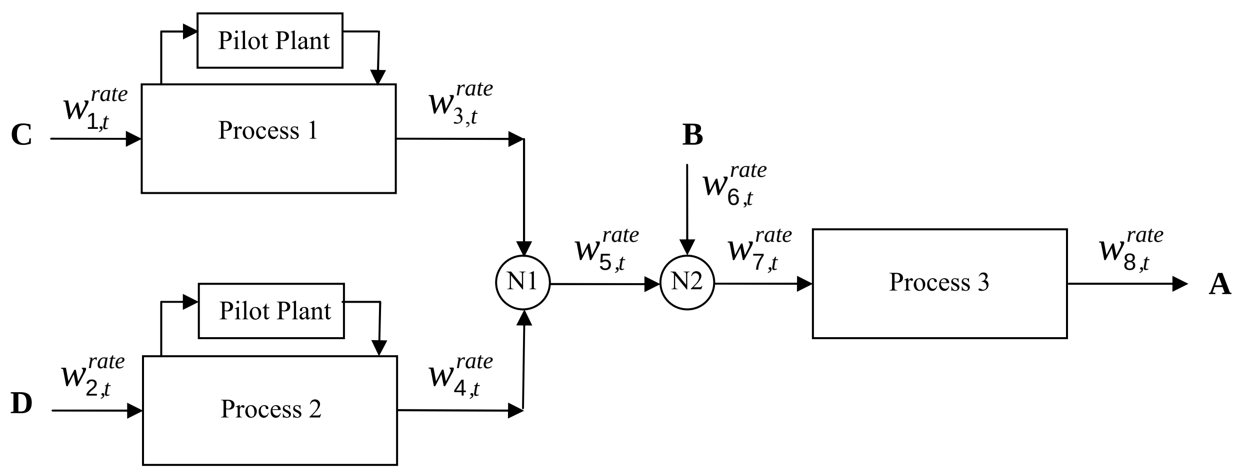

Figure 8: Schematic representation of process networks problem.

In Figure 8, process 3 is already operational with an existing capacity of 3 tons/day and a known yield of 70%. For the sake of simplicity, demand, which is the only exogenous uncertainty in the problem, is assumed to be known. Therefore, we only focus on the endogenous uncertainty and gradual uncertainty resolution. All possible realizations for process 1 at step 3 are 69, 73, 77 and 81% where only 69 and 81% are realizable in step 2 of uncertainty resolution (Figure 9). Similarly for process 2, 60 and 90% are two realizations in step 2 with 60, 70, 80 and 90% are possible realizations at step 3 (Figure 10). It is obvious from the parameters in the data that the only difference between the two different technologies is the variance of yield distributions. Although they possess the same mean value, 75%, process 2 has a higher variance than process 1. We also assume that the transition probabilities are identical in the scenario trees and the probability of each scenario is the equivalent. The scenarios generated by using these trees are shown in Table 1. All other parameter values used in the example problem are given in Table 2.

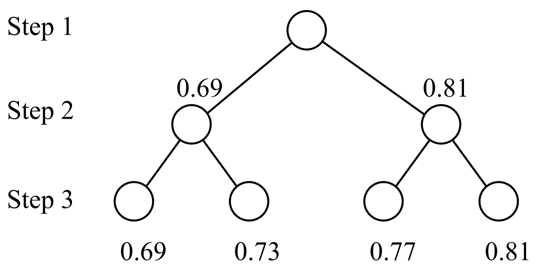

Figure 9: Gradual uncertainty resolution for process 1.

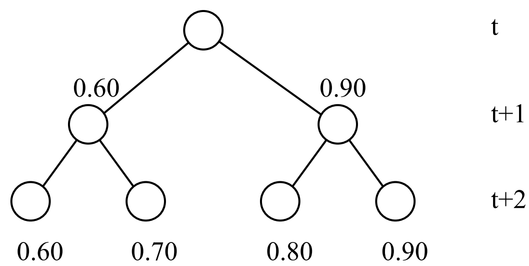

Figure 10: Gradual uncertainty resolution for process 2.

Table 1: Yield of each process in each scenario.

Scenario	**S1**	**S2**	**s3**	**s4**	**S5**	**s6**	**s7**	**s8**	**s9**	**s10**	**s11**	**s12**	**s13**	**s14**	**s15**	**s16**
Process 1	0.69	0.69	0.69	0.69	0.73	0.73	0.73	0.73	0.77	0.77	0.77	0.77	0.81	0.81	0.81	0.81
Process 2	0.60	0.70	0.80	0.90	0.60	0.70	0.80	0.90	0.60	0.70	0.80	0.90	0.60	0.70	0.80	0.90

Table 2: Values of the parameters used in the example.

	Costs				**Expansion**	
	Fixed Expansion ($ x 10^6 / expansion)	**Variable Expansion ($x 10^6 / ton)**	**Fixed Operating ($x 10^6 /year)**	**Pilot Plant ($x 10^6 /installation)**	**Max Limit (tons/day)**	**Min Limit (tons/day)**
Process 1	1.5	0.3	0.2	0.1	10	1
Process 2	1.5	0.3	0.2	0.1	10	1
Process 3	1.5	0.3	0.2	NA	10	1

Given the above instance of process networks in Figure 8, the optimization model for this instance is given in Appendix B.

5. DISJUNCTIVE PROGRAMMING MODEL

The general model for synthesis of process networks under uncertainty over T time periods is presented in this section. The general model optimizes expansion, operation of processes and investment decisions for pilot plants in a process network to maximize the expected net present value. The proposed mixed-integer/disjunctive programming model will be reformulated as a mixed integer linear program. The variables in the model can be classified as, decision variables, state variables and recourse variables. Decision variables are related to decisions that are taken at the beginning of each time period t and scenario s (e.g. expansion, pilot plant, operation, quantity of expansion and input flow rates). Recourse variables are decisions taken at the end of each period to satisfy feasibility (e.g. inventory levels, sales and purchase of final products). State variables are the variables that are automatically calculated when decision variables are selected (e.g. output flow rate from a process). The sequence of events is as follows. Decision variables are implemented at the beginning of time period t. This is followed by the gradual resolution of endogenous uncertainty. State variables are calculated automatically when the decision variables are set and the uncertain parameters are revealed. At the end of the period, recourse action is taken to satisfy feasibility.

It should be noted that in some scenarios, for a given set of decision variables and some values of uncertain parameters, we may end up with an infeasible solution. An example of such a situation will be discussed in the expected value solution in the results section. This infeasibility of decision variables does not cause difficulties in the Lagrangean decomposition because the model is decomposed into scenarios and solution of each scenario gives feasible solution for that scenario.

There are two criteria that are taken into consideration for selecting the variable names. The first one is a variable can be binary or continuous, and the second is a variable can be a control, state or recourse variable. The binary decision variables are denoted by y, whereas continuous variables are denoted by w or x. If a continuous variable is a control or a state variable w, is used, whereas if it is recourse variable x, is used. Based on this explanation, we define the sets, indices, variables and parameters for the process network as shown below:

Sets:

I : Processes in the process network

IU : Processes with uncertain yield in the process network

J : Nodes in the process network

K : Streams in the process network

S : Possible scenarios

T : Periods in time horizon

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

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

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

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

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

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

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

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

$M(i,s,s')$: Scenario pairs, s and s', which cannot be differentiated in the second step of uncertainty reduction after operating or installing pilot plant for process i

$STEP$: Steps for gradual uncertainty resolution

Indices:

i : Process in set I

j : Node in set J

k : Stream in set K

t,τ : Time period

s,s' : Scenario in set S

l : step in set $STEP$

Binary variables (or equivalent Boolean variables):

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

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

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

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

Continuous variables:

$w_{i,t}^{QE,s}$: Capacity expansion of process i in time period t, scenario s

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

$w_{k,t}^{rate,s}$: Flow rate of stream k in time period t, scenario s

$w_t^{inv,s}$: Inventory of final product at the end of time period t, scenario s

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

$x_t^{purch,s}$: Amount of purchases of final product in time period t, scenario s

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

ϕ : Expected net present value

Parameters:

p^s : Probability of scenario s

α_t : Purchase price for final product in time period t

β_t : Sales price for final product in time period t

γ_t : Cost of maintaining inventory of final product in period t

δ_t : Duration of time period t (~6 months)

$\theta_{i,l}^s$: Yield of process i in step l and scenario s

d_t : Demand for final product in time period t, scenario s

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

$VO_{j,t}$: Variable operating cost corresponding to $y_{k,t}^{rate,s}$

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

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

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

$FIPP_{i,t}$: Fixed investment cost for pilot plant for process i in time period t

$U_{(.)}^{(.)}$: Upper bounds

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

Based on the above definitions, the optimization model (P) for the synthesis of process networks is as follows.

Equation (1) represents the expected net present value which is to be maximized over the set of scenarios S.

$$(P) \qquad \phi = \sum_{s \in S} p^s NPV^s \tag{1}$$

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

$$\begin{aligned} NPV^s = &-\sum_{t\in T}\sum_{i\in I}\left(FE_{i,t}\,y_{i,t}^{\exp,s} + VE_{i,t}\,w_{i,t}^{QE,s}\right) - \sum_{t\in T}\sum_{i\in I} FO_{i,t}\,y_{i,t}^{oper,s} \\ &-\sum_{t\in T}\sum_{k\in K} VO_{k,t}\,w_{k,t}^{rate,s} - \sum_{t\in T}\left(\alpha_t x_t^{purch,s} - \beta_t x_t^{sales,s} + \gamma_t w_t^{inv,s}\right) \end{aligned} \qquad \forall s \in S \tag{2}$$

Equation (3) represents the input-output relationships for the processes in terms of the uncertain yields $\theta_{i,step}^{s}$ at each time period and scenario.

$$\sum_{k\in OP(i)} w_{k,t}^{rate,s} = \theta_{i,l}^{s} \sum_{k\in IP(i)} w_{k,t}^{rate,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \tag{3}$$

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

$$\sum_{k\in ON(j)} w_{k,t}^{rate,s} = \sum_{k\in IN(j)} w_{k,t}^{rate,s} \qquad \forall j \in J, \forall s \in S, \forall t \in T \tag{4}$$

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

$$w_t^{inv,s} = w_{t-1}^{inv,s} + \left(\sum_{k\in F} w_{k,t}^{rate,s} + x_t^{purch,s} - x_t^{sales,s}\right)\delta t \qquad \forall s \in S, \forall t \in T \tag{5}$$

Constraint (6) forces the sales to satisfy demand exactly for each time period and scenario,

$$x_t^{sales,s} = d_t \qquad \forall s \in S, \forall t \in T \qquad (6)$$

while constraint (7) restricts the outflow from each process by the capacity of the corresponding process.

$$\sum_{k \in OP(i)} w_{k,t}^{rate,s} \leq w_{i,t}^{cap,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad (7)$$

The capacities of every process in time period t and scenario s is computed by the equality constraint (8).

$$w_{i,t}^{cap,s} = w_{i,t-1}^{cap,s} + w_{i,t}^{QE,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad (8)$$

Upper and lower bounds for capacity expansions are calculated from constraint (9),

$$L_i^{QE} y_{i,t}^{\exp,s} \leq w_{i,t}^{QE,s} \leq U_i^{QE} y_{i,t}^{\exp,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad (9)$$

while constraint (10) represents the upper and lower bounds for outflows when the process is operating.

$$L_i^{\inf low} y_{i,t}^{oper,s} \leq \sum_{k \in OP(i)} w_{k,t}^{rate,s} \leq U_i^{\inf low} y_{i,t}^{oper,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad (10)$$

Logical constraint (11) states that operating a process for a given scenario s at time period t requires an expansion at any period before t, i.e. $\tau = 1,...,t$.

$$y_{i,t}^{oper,s} \Rightarrow \bigvee_{\tau=1}^{t} y_{i,\tau}^{\exp,s} \qquad \forall i \in IU, \forall s \in S, \forall t \in T \qquad (11)$$

Constraint (12) forces a process to be operated in time period t if its capacity is expanded in that time period.

$$y_{i,t}^{\exp,s} \Rightarrow y_{i,t}^{oper,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad (12)$$

Non-anticipativity constraints for the first period decisions, which define the root node of the scenario tree, are given by,

$$y_{i,1}^{oper,s} = y_{i,1}^{oper,s'} \qquad \forall i \in I, \forall (s,s') \in S, s < s' \qquad (13)$$

$$y_{i,1}^{pilot,s} = y_{i,1}^{pilot,s'} \qquad \forall i \in IU, \forall (s,s') \in S, s < s' \qquad (14)$$

$$y_{i,1}^{\exp,s} = y_{i,1}^{\exp,s'} \qquad \forall i \in I, \forall (s,s') \in S, s < s' \qquad (15)$$

$$w_{i,1}^{QE,s} = w_{i,1}^{QE,s'} \qquad \forall i \in I, \forall (s,s') \in S, s < s' \qquad (16)$$

$$w_{k,1}^{rate,s} = w_{k,1}^{rate,s'} \qquad \forall k \in DK, \forall (s,s') \in S, s < s' \qquad (17)$$

Constraint (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.

$$\sum_{\tau=1}^{T} y_{i,\tau}^{pilot,s} \le 1 \qquad \forall i \in IU, \forall s \in S \qquad (18)$$

The logic constraint (19) states that for a given scenario s, if there has been an expansion in the process until time period t, then there is no need for a pilot plant for a process having uncertain yield at time period t, scenario s.

$$\bigvee_{\tau=1}^{t} y_{i,\tau}^{\exp,s} \Rightarrow \neg y_{i,t}^{pilot,s} \qquad \forall i \in IU, \forall s \in S, \forall t \in T \qquad (19)$$

Also, for a given process having uncertain yield, if there is a pilot plant installation at time period t, no expansion can be made in the same period.

$$y_{i,t}^{pilot,s} \Rightarrow \neg y_{i,t}^{\exp,s} \qquad \forall i \in IU, \forall s \in S, \forall t \in T \qquad (20)$$

Logic constraint (21) states that, s and s' can be indistinguishable in two ways (see Figure 3). The first one is that there is no expansion or pilot plant installation in a process that differentiates s and s'. The second way is that process i, that differentiates the scenarios s and s', is operated for just one period or a pilot plant for process i is installed, and s and s' are two scenarios such that they cannot be differentiated in the second step of uncertainty reduction (handled by $(i,s,s') \in M(i,s,s')$). Other than these two, scenarios s and s' will be distinguishable. This logic equivalence is actually a reduced version of the original logic constraint. Appendix A provides a proof for the reduction of the original logic constraint.

$$Z_t^{s,s'} \Leftrightarrow \bigwedge_{i \in D(s,s')}^{I} \left\{ \left[\bigwedge_{\tau=1}^{t} \neg (y_{i,\tau}^{\exp,s} \vee y_{i,\tau}^{pilot,s}) \right] \underline{\vee} \left[\left(\bigvee_{\tau=1}^{t} (y_{i,\tau}^{oper,s} \underline{\vee} y_{i,\tau}^{pilot,s}) \right) \wedge \left((i,s,s') \in M(i,s,s') \right) \right] \right\}$$

$$\forall (s,s'), s < s', \forall t \in T \qquad (21)$$

Finally, the disjunctive constraint (22) shows the decisions that are going to be the same for time period t and t+1 if s and s' are indistinguishable at time period t.

$$\begin{bmatrix} Z_t^{s,s'} & \\ x_t^{purch,s} = x_t^{purch,s'} & \\ x_t^{sales,s} = x_t^{sales,s'} & \\ y_{i,t+1}^{oper,s} = y_{i,t+1}^{oper,s'} & if\ t \le T-1, i \in I \\ y_{i,t+1}^{\exp,s} = y_{i,t+1}^{\exp,s'} & if\ t \le T-1, i \in I \\ y_{i,t+1}^{pilot,s} = y_{i,t+1}^{pilot,s'} & if\ t \le T-1, i \in IU \\ w_{i,t+1}^{QE,s} = w_{i,t+1}^{QE,s'} & if\ t \le T-1, i \in I \\ w_{k,t+1}^{rate,s} = w_{k,t+1}^{rate,s'} & if\ t \le T-1, k \in DK \end{bmatrix} \vee \left[\neg Z_t^{s,s'} \right] \quad \forall (s,s'), s < s', \forall t \in T \qquad (22)$$

Appendix B presents the specific formulation for the motivating example.

It should be noted that model (P), given by equations (1)-(22) corresponds to a mixed integer/disjunctive programming model. The most direct way of solving this problem is to convert it into an MILP model by converting the logic propositions into linear inequalities and disjunctions into mixed-integer constraints (e.g. see Raman and Grossmann, 1994).

6. SOLUTION METHOD

Model (P) is coupled through the non-anticipativity constraints that relate the various scenarios. In the proposed algorithm, the upper bounds at each node are generated by solving a Lagrangean dual problem. This Lagrangean problem is obtained by relaxing logic constraint (21), disjunction (22) and by dualizing the first period non-anticipativity constraints (13)-(17), which involves transferring them into the objective function multiplied by the corresponding Lagrange multipliers (λ). For fixed values of these multipliers, the problem can be decomposed into independent Lagrangean MILP sub-problems for each scenario.

At the root node the formulation of the Lagrangean dual is as follows,

$$(P_{LR}) \quad \max \quad \phi_{LR}\left(\lambda_{oper,i,t}^{s,s'}, \lambda_{pilot,i,t}^{s,s'}, \lambda_{exp,i,t}^{s,s'}, \lambda_{QE,i,t}^{s,s'}, \lambda_{rate,k,t}^{s,s'}\right) = \sum_{s \in S} p^s NPV^s$$

$$\begin{aligned} NPV^s = & -\sum_{t \in T}\sum_{i \in I}\left(FE_{i,t} y_{i,t}^{exp,s} + VE_{i,t} w_{i,t}^{QE,s}\right) - \sum_{t \in T}\sum_{i \in I} FO_{i,t} y_{i,t}^{oper,s} \\ & - \sum_{t \in T}\sum_{k \in K} VO_{k,t} w_{k,t}^{rate,s} - \sum_{t \in T}\left(\alpha_t x_t^{purch,s} - \beta_t x_t^{sales,s} + \gamma_t w_t^{inv,s}\right) \\ & + \sum_{\substack{s,s' \\ s<s'}}\left(\sum_{i \in I} \lambda_{oper,i,1}^{s,s'}\left(y_{i,1}^{oper,s} - y_{i,1}^{oper,s'}\right) + \sum_{i \in IU} \lambda_{pilot,i,1}^{s,s'}\left(y_{i,1}^{pilot,s} - y_{i,1}^{pilot,s'}\right) + \sum_{i \in I} \lambda_{exp,i,1}^{s,s'}\left(y_{i,1}^{exp,s} - y_{i,1}^{exp,s'}\right)\right. \\ & \left. + \sum_{i \in I} \lambda_{QE,i,1}^{s,s'}\left(w_{i,1}^{QE,s} - w_{i,1}^{QE,s'}\right) + \sum_{k \in DK} \lambda_{rate,k,1}^{s,s'}\left(w_{k,1}^{rate,s} - w_{k,1}^{rate,s'}\right)\right) \end{aligned}$$

s.t. (3) - (12)

and

(18) - (20)

The parameters $\lambda_{oper,i,1}^{s,s'}$, $\lambda_{pilot,i,1}^{s,s'}$, $\lambda_{\exp,i,1}^{s,s'}$, $\lambda_{QE,i,1}^{s,s'}$ and $\lambda_{rate,k,1}^{s,s'}$ represent the Lagrange multipliers corresponding to constraints $y_{i,1}^{oper,s} = y_{i,1}^{oper,s'}$, $y_{i,1}^{pilot,s} = y_{i,1}^{pilot,s'}$, $y_{i,1}^{\exp,s} = y_{i,1}^{\exp,s'}$, $w_{i,1}^{QE,s} = w_{i,1}^{QE,s'}$ and $w_{k,1}^{rate,s} = w_{k,1}^{rate,s'}$, respectively. The logic constraints such as (11)-(12), (19)-(20) can be written as linear inequalities (see Williams, 1999). We should note that model (P_{LR}) is the Lagrangean relaxation of the original model (P) in which the logic constraint (21) and the disjunction (22) have been removed.

Since the MILP model $\left(P_{LR}\right)$ is a relaxation of the original model for any fixed values of the Lagrange multipliers, its solution yields an upper bound. That is,

$$\phi \leq \phi_{LR}\left(\lambda^{s,s'}_{oper,i,t}, \lambda^{s,s'}_{pilot,i,t}, \lambda^{s,s'}_{exp,i,t}, \lambda^{s,s'}_{QE,i,t}, \lambda^{s,s'}_{rate,k,t}\right) \qquad \forall \lambda^{s,s'}_{oper,i,t}, \forall \lambda^{s,s'}_{pilot,i,t}, \forall \lambda^{s,s'}_{exp,i,t}, \forall \lambda^{s,s'}_{QE,i,t}, \forall \lambda^{s,s'}_{rate,k,t}$$

In order to find the tightest upper bound generated by model P_{LR}, we consider the minimization of the Lagrangean dual which is defined as,

$$\text{Min } \phi_{LD} = \min_{\lambda} \ \phi_{LR}\left(\lambda^{s,s'}_{oper,i,t}, \lambda^{s,s'}_{pilot,i,t}, \lambda^{s,s'}_{exp,i,t}, \lambda^{s,s'}_{QE,i,t}, \lambda^{s,s'}_{rate,k,t}\right)$$

In the above problem we search for the lowest upper bound in the space of multipliers. This search is performed by the subgradient method proposed by Fisher (1985).

In general the solution of the Lagrangean dual may not satisfy the first period non-anticipaticity constraints (13)-(17) and the relaxed disjunction (22). In this case, new branches are generated from the current node by considering the violations in the relaxed disjunctions and constraints. Upper bounds at each node are generated by solving a similar Lagrangean dual problem. Model $\left(P_n\right)$ represents the problem to be solved at each node,

$$\left(P_n\right) \quad \max \quad \phi_n = \sum_{s \in S} p^s NPV^s$$

$$NPV^s = -\sum_{t \in T}\sum_{i \in I}\left(FE_{i,t} y^{\exp,s}_{i,t} + VE_{i,t} w^{QE,s}_{i,t}\right) - \sum_{t \in T}\sum_{i \in I} FO_{i,t} y^{oper,s}_{i,t} - \sum_{t \in T}\sum_{k \in K} VO_{k,t} w^{rate,s}_{k,t} - \sum_{t \in T}\left(\alpha_t x^{purch,s}_t - \beta_t x^{sales,s}_t + \gamma_t w^{inv,s}_t\right)$$

s.t. (3) - (12)

$$\left.\begin{array}{lll} y^{oper,s}_{i,t} = y^{oper,s'}_{i,t} & \forall i \\ y^{pilot,s}_{i,t} = y^{pilot,s'}_{i,t} & \forall i \\ y^{\exp,s}_{i,t} = y^{\exp,s'}_{i,t} & \forall i \\ w^{QE,s}_{i,t} = w^{QE,s'}_{i,t} & \forall i \\ w^{rate,s}_{k,t} = w^{rate,s'}_{k,t} & \forall k \\ x^{purch,s}_t = x^{purch,s'}_t \\ x^{sales,s}_t = x^{sales,s'}_t \end{array}\right\} \quad \forall (s,s',t) \in N_E \qquad (23)$$

and

(18) - (20)

The set N_E consists of tuples (s,s',t) for which non-anticipativity constraints apply. Unlike the first period non-anticipativity constraints that are added to the model at the root node, non-anticipativity constraints coming from the relaxed disjunction are added to the model depending on the optimal values of variables at the parent node of the branch and bound tree. If the optimal values at the parent node imply $Z_t^{s,s'}$ to be true by logic constraint (21), the non-anticipativity constraints in the disjunction are added to the set N_E. The Lagrangean relaxation of the model (P_n) at each node is generated by dualizing the equalities in (23) and transferring them to the objective function with Lagrange multipliers.

The lower bound at each node is found by a heuristic, taking the solution found by the upper bound generation as an input. We fix all binary variables for installation, pilot plant and some of the binary variables for operation for processes such that the first period (13)-(17) and the disjunctive non-anticipativity constraints (21)-(22) are satisfied. With fixed decisions we solve the model (P) as an MILP. Note that it is possible to obtain an infeasible solution by violating constraint (11). In order to avoid this type of infeasible results, constraint (11) is also considered before fixing the binary variables for operation.

In order to decide the value of $y_{i,t}^{(.),s}$ to fix for the indistinguishable scenarios, we find an α value by calculating the probability weighted average of the infeasible solution for process i and time period t as $\alpha = \sum_{j=1}^{K} \hat{y}_{i,t}^{(.),s_j} p^{s_j} - \sum_{j=1}^{K} \left(1 - \hat{y}_{i,t}^{(.),s_j}\right) p^{s_j}$, where $\{s_1, s_2, ..., s_K\}$ is the set of indistinguishable scenarios found by logic constraint (21) at time period t. If α has a value greater than or equal to zero then we set the variable to 1 otherwise it is set to 0.

Based on the above definitions of the models, the proposed algorithm is presented below. In the algorithm, $\mathcal{P}$ denotes the list of open nodes each having an upper bound ϕ^{UB} found by the Lagrangean dual problem, while ϕ^{LB} represents the objective value of the best feasible solution obtained.

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

Step 2: *Termination:* If $\mathcal{P} = \emptyset$, stop. Current best solution is optimal.

Step 3: *Node selection:* Select and delete a problem (P_m) from $\mathscr{P}$. Using the logic constraint and the fixed variables at the current node, add the conditional non-anticipativity constraints to the set N_E in (P_m) as shown in model (P_n).

Step 4: *Solution:* Solve the Lagrangean relaxation of the problem (P_m) to obtain solution $(\hat{y},\hat{w},\hat{x})$ with objective function value $\hat{\phi}(P_m)$. If the solution is infeasible, set $\phi^{UB} = \hat{\phi}(P_m)$, otherwise the solution is feasible, set $\overline{\phi}(P_m) = \hat{\phi}(P_m)$ and go to Step 6.

Step 5: *Bounding:* If $\phi^{LB} \geq \phi^{UB}$, go to Step 2. Otherwise, find a heuristic solution based on $(\hat{y},\hat{w},\hat{x})$ to generate a feasible solution $(\overline{y},\overline{w},\overline{x})$ with objective value $\overline{\phi}(P_m)$

Step 6: *Update bound:* Update the lower bound by $\phi^{LB} = max\{\phi^{LB},\overline{\phi}(P_m)\}$. Delete from $\mathscr{P}$ all problems P' with $\hat{\phi}(P') \leq \phi^{LB}$. If solution $(\hat{y},\hat{w},\hat{x})$ found in Step 4 is feasible, go to Step 2.

Step 7: *Branching:* We branch on the dualized constraints which are violated by $(\hat{y},\hat{w},\hat{x})$, the solution of the relaxed problem(P_m). The branching strategy depends on whether the variables involved are binary or continuous. If the variables are binary, two branches divide the feasible region into two, where one of the regions is restricted to $y_{i,t}^{(.),s} = y_{i,t}^{(.),s'} = 0$, and the other $y_{i,t}^{(.),s} = y_{i,t}^{(.),s'} = 1$. On the other hand, if the variables are continuous, two branches divide the feasible region into two, where one of the regions has to satisfy $w_{.,t}^{(.),s} = w_{.,t}^{(.),s'} \leq \widetilde{w}_{.,t}$ and $w_{.,t}^{(.),s} = w_{.,t}^{(.),s'} \geq \widetilde{w}_{.,t}$ has to hold in the other branch. $\widetilde{w}_{.,t}$ is calculated by the weighted average of the two variables, $\widetilde{w}_{.,t} = \dfrac{p^s w_{.,t}^{(.),s} + p^{s'} w_{.,t}^{(.),s'}}{p^s + p^{s'}}$.

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, $y_{i,1}^{(.),s_1} = y_{i,1}^{(.),s_2} = ... = y_{i,1}^{(.),s_k}$. The branching can then be performed so that the entire expression is restricted to 0 (i.e. $y_{i,1}^{(.),s_1} = y_{i,1}^{(.),s_2} = ... = y_{i,1}^{(.),s_k} = 0$) or to 1 (i.e. $y_{i,1}^{(.),s_1} = y_{i,1}^{(.),s_2} = ... = y_{i,1}^{(.),s_k} = 1$).

7. IMPLEMENTATION

We have implemented the proposed algorithm in C++ using ILOG Concert Technology (ILOG, 2005b). In this implementation, there is a basic branch and bound file which controls the search in this algorithm. The file manages the list of active nodes to be explored, prunes the nodes that are guaranteed to have suboptimal solutions, and creates children nodes depending on the unsatisfied non-anticipativity constraints. All the data for each node in the branch and bound tree are stored in a separate data file on the hard disk so there is no problem of running out of memory even if the size of the tree becomes very large.

In every node, an upper bound and a lower bound are calculated. The upper bound is generated by solving (P_n) and updating the Lagrange multipliers using subgradient optimization, whereas the lower bounds are found by modifying the infeasible solution generated by the upper bounding scheme. When performing the first iteration at a given node, the initial values of Lagrange multipliers are taken from the ones at the last iteration of the parent node. The step size used for finding Lagrange multipliers is halved each time the upper bound does not improve for a specified number of iterations. In this case, starting with the best lower bound instead of the last iteration improves the lower bounds drastically. For the lower bound, the binary variables are fixed according to the rule explained in the algorithm, and the model is solved in full space with these fixed variables.

We should note that this code can be used for solving similar types of problems by customizing the lower bounding scheme to take advantage of the specifics of the problem.

8. RESULTS

In this section, we present results for the motivating example given in section 4 to show the effectiveness of the proposed solution algorithm. All the results presented in this section have been obtained on a Pentium-IV, 3.20 GHz Windows machine. Also, we employed ILOG CPLEX 9.1 and Concert Technology 2.1.

Table 3: Model size in full space.

	Motivating Example (16 Scenarios)
0-1 Variables	7360
Continuous Variables	8841
Constraints	85139

Table 4: Results for the solution of full space model.

	Motivating Example (16 Scenarios)				
Elapsed time (seconds)	10,000	20,000	50,000	100,000	250,000
Lower Bound ($ x 10^6)	61.55	61.55	61.55	61.55	61.55
Optimality gap (%)	281.68	281.44	277.16	273.11	270.34

Table 3 and 4 show the size of the model and the results obtained by ILOG CPLEX after running the full space model for the specified amount of time. The results show that CPLEX finds a lower bound (feasible solution) with an expected net present value of $\$61.55 \times 10^6$ relatively early, but it can not improve it and close the corresponding integrality gap. The size of the full space model and the unsatisfactory results in trying to optimize the model clearly motivates the need of the proposed duality-based branch and bound algorithm.

Figure 11 and 12 shows the best feasible solution found by CPLEX after running the full space model for 250,000 seconds. This solution proposes expanding the capacity of process 2 to 10 tons/day and then if the yield occurs as 60%, it expands process 1 to 7.87 tons/day and process 3 by 8.43 tons/day (see Figure 11). If the yield of the process 2 is higher than 60%, then it does not install process 1 but it expands the current capacity of process 3 by 7.48 (see Figure 12).

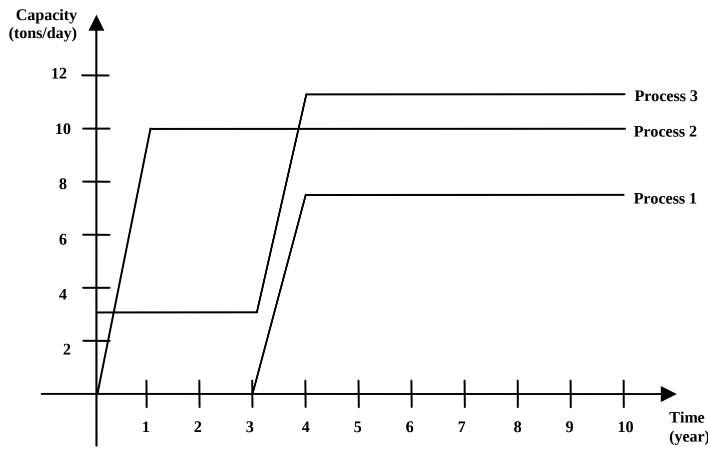

Figure 11: Best feasible solution proposed by CPLEX after 250,000 seconds
(if the yield of process 2 is 60%).

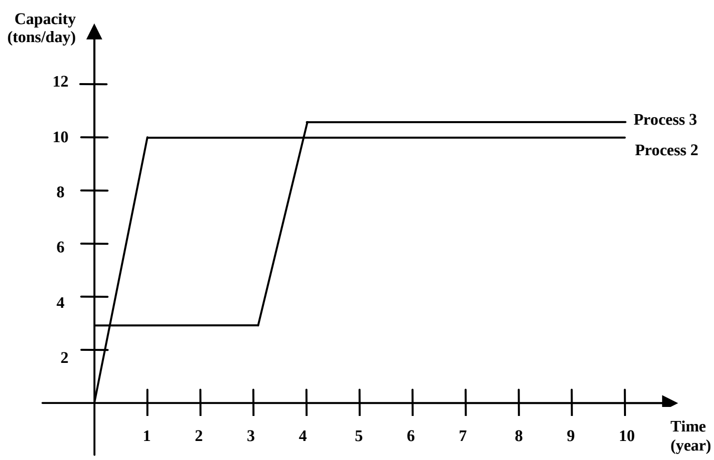

Figure 12: Best feasible solution proposed by CPLEX after 250,000 seconds
(if the yield of process 2 is higher than 60%).

Figure 13 shows the branch and bound tree generated by the proposed algorithm which found the optimal or near optimal solution to have an expected net present value of 80.14 x 10^6. The root node is denoted by 0 and the rest of the nodes are number according to the order of exploration. At the root node, the Lagrangean problem yields an upper bound of 88.48 x 10^6 after 18 iterations of subgradient optimization and the heuristic for finding a feasible solution yields a lower bound of 76.68 x 10^6. In the optimal solution of the Lagrangean problem at the root node, the 1st stage non-anticipativity constraints for $y_{1,1}^{\exp,s}$ are violated. Therefore, the feasible solution is separated into two, where $y_{1,1}^{\exp,s}=0$ or $y_{1,1}^{\exp,s}=1$, which are represented by nodes 1 and 2.

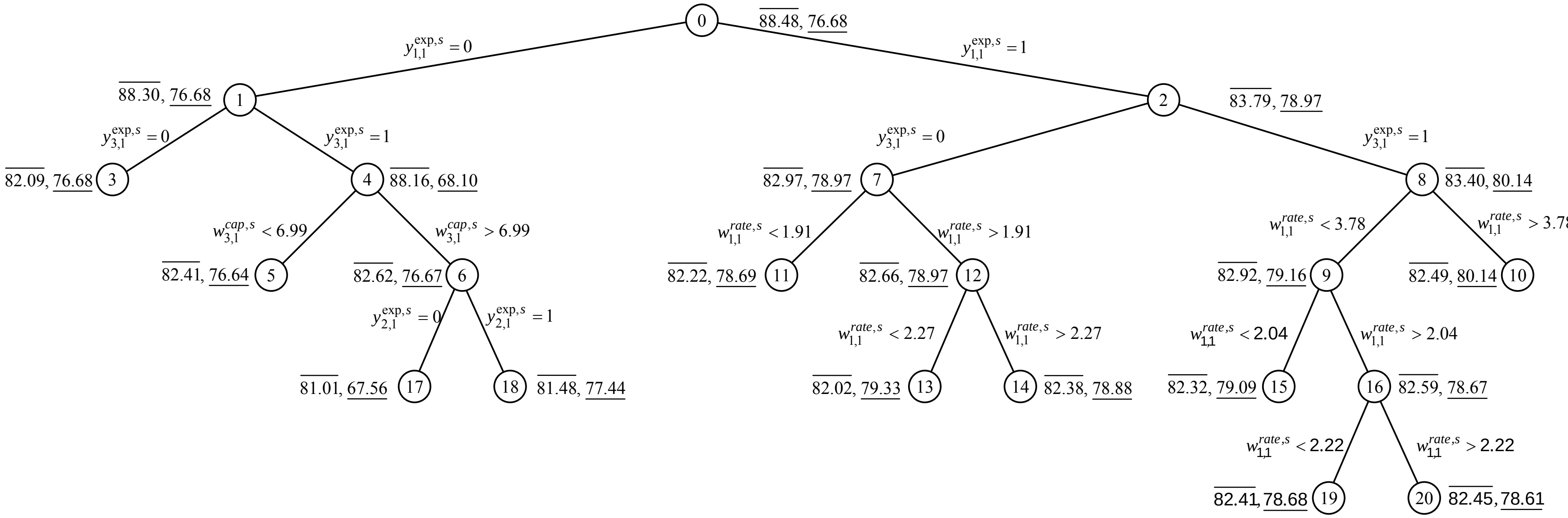

Figure 13: Resulting Branch and Bound tree for motivating example.

The proposed solution is guaranteed to be within 3% integrality gap and the proposed branch and bound algorithm required 36,047 seconds. The best feasible solution was found after 28,400 seconds and then the remaining time is spent to reduce the integrality gap. Table 5 compares the two snapshots of the iterations, where one of them after the best feasible solution is found and the other one is after the termination of the proposed algorithm.

Table 5: Gap at two points of proposed algorithm.

	Time (seconds)	Gap (%)
After best feasible solution found	28400	9.9
After termination of the algorithm	36047	3

Table 6 shows the best feasible solution with objective 80.14 x 10^6 found by the proposed branch and bound method as shown in Figure 13. The best solution proposes expanding the capacity of process 1 and 3 at the first time period by 10.0 and 7.0 tons/day respectively and further expand the process 1 capacity depending on the outcome of the yield. If the yield is low (scenarios 1 to 8), it needs larger capacity expansion of 4.49 (see Figure 14), compared to the smaller capacity of 2.98 tons/day (see Figure 15) for scenarios where the yield is higher (scenarios 9 to 16). Also, the solution proposes to operate processes 1 and 3 all the time periods except the second one. Table 6 shows the capacity expansions at each time period. The capacity of each process at time t can be calculated by summing up the capacity expansion and considering the initial capacities.

Table 6: Best feasible solution found.

Time	s1	s2	s3	s4	s5	s6	s7	s8	s9	s10	s11	s12	s13	s14	s15	s16
1	$w_{1,1}^{QE,s}=10.0, w_{3,1}^{QE,s}=7.00, y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$															
2	--								--							
3	$w_{1,1}^{QE,s}=4.49, y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$								$w_{1,1}^{QE,s}=2.98, y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$							
4	$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$			
5	$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$			
6	$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$			
7	$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$			
8	$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$			
9	$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$			
10	$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$				$y_{1,1}^{oper,s}=1, y_{3,1}^{oper,s}=1$			
ENPV	\$ 80.14 x 10^6															

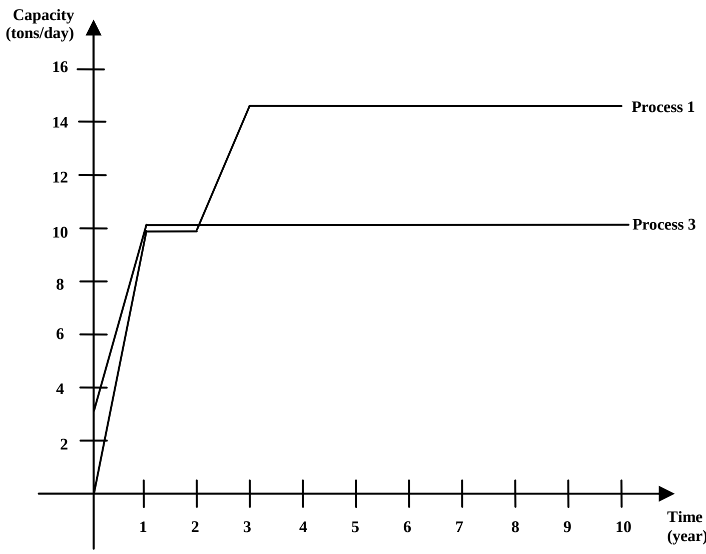

Figure 14: Solution by proposed Branch and Bound Algorithm
(if the yield of process 1 is 69%, i.e. for scenarios s1 - s8).

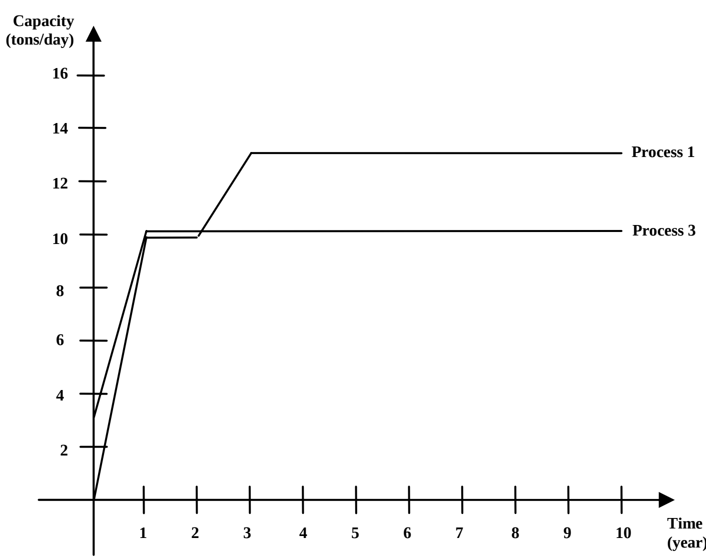

Figure 15: Solution by proposed Branch and Bound Algorithm
(if the yield of process 1 is 81%, i.e. for scenarios s9 – s16).

If we compare the best feasible solution of the proposed Branch and Bound and the full space, it is clear that both solutions take the variation in the yields into consideration as expected. The branch and bound solution, with a higher objective, refrains from investing in process 2 because the uncertainty of the yield is higher than process 1. Furthermore, it proposes a more conservative approach than full space by expanding the capacity of the process with smaller yield uncertainty in the further time periods. Also, it proposes expanding the current capacity of process 3 from the beginning so that the new processes can work to full capacity during entire project horizon.

It is interesting to note that the proposed algorithm generates very good feasible solutions at the root node and at the first two nodes (Node 1 and 2) as shown in Table 7. In fact, the lower bound of $76.68 x 10^6 obtained at the root node at nearly 5172 seconds is much better than the feasible solution of $61.55 x 10^6 found by running the full space model for 250,000 seconds with CPLEX.

Table 7: Comparison of results found at the root node, node 1 and 2 and the full space model.

	Solution time (seconds)	Upper Bound ($ x 10^6)	Lower Bound ($ x 10^6)	Gap (%)
Only Root Node	5172	88.48	76.68	15.38
Node 1	4278	88.30	76.68	15.15
Node 2	620	83.79	78.97	6.10
Full space	250,000	229.62	61.55	269.13

Table 8 provides the results under perfect information (i.e. wait-and-see solution). For every scenario, the set of decisions are found under the assumption that the decision maker has all the actual values of the uncertain parameters beforehand. As decisions for each scenario will be the best that the decision maker can take independent of other scenarios, the probability weighted average of these solutions will also yield an upper bound for the problem. This is actually the same solution found in the first iteration at the root node, when the Lagrange multipliers of the 1st period non-anticipativity constraints are set to zero.

Table 8: Results under perfect information.

Scenario	Objective Value ($ x 10^6)
Scenario 1	69.51
Scenario 2	70.80
Scenario 3	92.31
Scenario 4	110.83
Scenario 5	78.02
Scenario 6	78.02
Scenario 7	92.31
Scenario 8	110.83
Scenario 9	86.56
Scenario 10	86.56
Scenario 11	92.31
Scenario 12	110.83
Scenario 13	93.43
Scenario 14	93.43
Scenario 15	93.43
Scenario 16	110.83
Expected Value	91.87

As a comparison to the multistage stochastic programming approach, one can try to solve the problem where the values of uncertain parameters are fixed to the expected values. This solution provides the set of decisions when the variation in uncertain parameters is not taken into account. For the motivating example, the yield values are fixed to the average yield, 75% for both processes. The decision variables found from this solution are fixed and the original problem (P) was solved in full space. Unfortunately, the fixed decision variables may not be feasible for some scenarios. For example, let us assume that we fix the input flow rates $w_{1,t}^{rate,s}$, $w_{2,t}^{rate,s}$ and $w_{6,t}^{rate,s}$ and the uncertain yield takes high possible value, say 81%. Then, the sum of the output flow rate from process 1 and the fixed flow rate of intermediate product, $w_{6,t}^{rate,s}$, can be greater than the capacity of process 3, making the solution infeasible. This type of infeasibilities make the expected value solution of very limited use.

As another comparison, a feasible, but not optimal solution which includes a pilot plant installation is presented. To find a feasible solution, the following argument was employed. If the variation in the yield of a process is high, then it may be wise to build a pilot plant first and then make decisions based on the outcomes of these decisions. We first install a pilot plant for the process 2 which has a higher variation in yield, and then if the yield turns out to be high, we install the actual plant, otherwise we install the process 1, which has a smaller variation. Table 9 shows the binary decisions to be fixed for this feasible solution. The empty cells in the table represent that no variable has been fixed. The indistinguishable scenario pairs are found by using logic constraint 21, and the columns of indistinguishable scenarios are merged for a simpler representation.

Table 9: A Feasible (not optimal) solution with installation of pilot plant.

<table>
<tr><th>Time</th><th>s1</th><th>s2</th><th>s5</th><th>s6</th><th>s9</th><th>s10</th><th>s13</th><th>s14</th><th>s3</th><th>s7</th><th>s11</th><th>S15</th><th>s4</th><th>s8</th><th>s12</th><th>s16</th></tr>
<tr><td>1</td><td colspan="16">$y_{2,1}^{pilot,s} = 1$, $y_{3,1}^{QE,s} = 7.47$, $y_{3,1}^{oper,s} = 1$</td></tr>
<tr><td>2</td><td colspan="8">$y_{1,2}^{\exp,s} = 10$, $y_{1,2}^{oper,s} = 1$, $y_{3,2}^{oper,s} = 1$</td><td colspan="8">$y_{2,2}^{\exp,s} = 10$, $y_{3,2}^{oper,s} = 1$</td></tr>
<tr><td>3</td><td colspan="4">$y_{1,3}^{oper,s} = 1$, $y_{1,3}^{oper,s} = 1$</td><td colspan="4">$y_{1,3}^{oper,s} = 1$, $y_{1,3}^{oper,s} = 1$</td><td colspan="4">$y_{2,3}^{oper,s} = 1$, $y_{3,3}^{oper,s} = 1$</td><td colspan="4">$y_{2,3}^{oper,s} = 1$, $y_{3,3}^{oper,s} = 1$</td></tr>
<tr><td>4</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="4">“</td><td colspan="4">“</td></tr>
<tr><td>5</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="4">“</td><td colspan="4">“</td></tr>
<tr><td>6</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="4">“</td><td colspan="4">“</td></tr>
<tr><td>7</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="4">“</td><td colspan="4">“</td></tr>
<tr><td>8</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="4">“</td><td colspan="4">“</td></tr>
<tr><td>9</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="4">“</td><td colspan="4">“</td></tr>
<tr><td>10</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="2">“</td><td colspan="4">“</td><td colspan="4">“</td></tr>
<tr><td>ENPV</td><td colspan="16">$ 77.28 x 10^6</td></tr>
</table>

This solution gives an expected net present value of $77.28 x 10^6 which is lower than the solution $80.14 x 10^6 found by proposed branch and bound algorithm. Finding the optimal solution after fixing these variables takes only about 106 seconds, and if we compare this result with the results in Table 4, this heuristic gives a better solution than proposed by running the full space problem with CPLEX for 250,000 seconds. On the other hand, this heuristic is not guaranteed to provide the optimal or even a feasible solution.

9. CONCLUSION

In this paper, we have presented a multistage stochastic programming model with decision dependent elements for the synthesis of process networks with time-varying uncertain yields. The proposed model considers possible investment strategies of pilot plants as a way of reducing uncertainties before making the major plant investments. In order to consider complex economic trade-offs involved in this problem, a stochastic programming model was proposed, which led to a mixed integer/disjunctive programming model. Since this model can only be solved with an LP-based branch and bound for smaller instances, we have proposed a duality-based branch and bound algorithm for solving the larger problems.

Implementation issues and results of an example were presented to illustrate the application of the proposed method. The size of the proposed model for the example presented is quite large (7360 binary variables, 8841 continuous variables and 85,139 constraints). Therefore the direct solution through CPLEX proved to be very expensive and can not find the optimal solution after 250,000 seconds. In contrast, the proposed method required 36,047 seconds to solve the problem with 3% optimality, finding good solutions early in the search. Although the preliminary results are encouraging, more extensive testing will be required to fully assess the effectiveness of the proposed method.

ACKNOWLEDGMENTS:

The authors acknowledge the financial support from ExxonMobil 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. Ph.D. thesis, University of Illinois at Urbana-Champaign.

Birge, J.R. , Louveaux, F. (1997). Introduction to stochastic programming. New York, NY: Springer.

Dias, M., 2002. Real Options in Real Life, MIT Seminar.

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., 2005. A class of stochastic program with decision dependent uncertainty. (Accepted for publication)

Grossmann, I.E., 2002. Review of non-linear mixed integer and disjunctive programming techniques. Optimization and Engineering 3, 227-252.

Held, H., D.L., Woodruff, 2005. Heuristics for Multi-Stage Interdiction of Stochastic Networks. Journal of Heuristics 11, 483-500.

ILOG, 2005a. ILOG CPLEX 9.1 User's Manual.

ILOG, 2005b. ILOG CPLEX and Concert C++ API 9.1 Reference Manual.

Jonsbraten, T.W., 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.

Luceno, Alberto, 1999. Discrete approximations to continuous univariate distributions – an alternative simulation. Journal of the Royal Statistical Society (Ser B) 61, 345-352.

Pflug, G. Ch., 1990. Online optimization of simulated markovian processes. Mathematics of Operations Research Res. 15, No.3, 381-395.

Raman, R., Grossmann, I.E., 1994. Modeling and computational techniques for logic-based integer programming. Computers and Chemical Engineering 18 (7), 563-578.

Ruszczynski, A., 1997. Decomposition methods in stochastic programming. Math. Programming (Ser. B) 79, 333-353.

Stensland, G., Tjostheim, D. B., 1991. Optimal decisions with reduction of uncertainty over time – an application to oil production. Stochastic Models and Option Values, Elsevier Science Publishers B.V, pp. 267-291.

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

Vishwanath, K., Peeta, S., Salman, S., 2004. Investing in the links of a stochastic network to minimize expected shortest path length (working paper).

H.P. Williams., Model Building in Mathematical Programming. John Wiley and Sons, New York, New York, 4th edition, 1999.

APPENDIX A

Equation (A1) can be reduced into equation (21) by using propositions 1 and 2.

$$Z_t^{s,s'} \Leftrightarrow \bigwedge_{i\in D(s,s')}^{I} \left\{ \left(\left[\bigwedge_{\tau=1}^{t} \neg (y_{i,\tau}^{\exp,s} \vee y_{i,\tau}^{pilot,s}) \right] \wedge \left[\bigwedge_{\tau=1}^{t} \neg (y_{i,\tau}^{\exp,s'} \vee y_{i,\tau}^{pilot,s'}) \right] \right) \right.$$

$$\left. \underline{\vee} \left[\left(\underline{\bigvee_{\tau=1}^{t}} (y_{i,\tau}^{oper,s} \underline{\vee} y_{i,\tau}^{pilot,s}) \right) \wedge \left(\underline{\bigvee_{\tau=1}^{t}} (y_{i,\tau}^{oper,s'} \underline{\vee} y_{i,\tau}^{pilot,s'}) \right) \wedge \left((i,s,s') \in M(i,s,s') \right) \right] \right\}$$

$$\forall (s,s'), s < s', \forall t \in T \qquad \text{(A1)}$$

Proposition 1:

$$\left[\bigwedge_{\tau=1}^{t} \neg (y_{i,\tau}^{\exp,s} \vee y_{i,\tau}^{pilot,s}) \right] \Rightarrow \left[\bigwedge_{\tau=1}^{t} \neg (y_{i,\tau}^{\exp,s'} \vee y_{i,\tau}^{pilot,s'}) \right] \quad where\ s < s' \qquad \text{(A2)}$$

Proof:

Let us assume $\left[\bigwedge_{\tau=1}^{t} \neg (y_{i,\tau}^{\exp,s} \vee y_{i,\tau}^{pilot,s}) \right]$ is true. Then, there is no expansion or pilot plant installation to process i until time period t in scenario s. So the expression $\left[\bigwedge_{\tau=1}^{\hat{t}} \neg (y_{i,\tau}^{\exp,s} \vee y_{i,\tau}^{pilot,s}) \right]$ should hold true for each $\hat{t} \le t$. At $\hat{t} = 1$, first period non-anticipativity constraints (eq. (13)-(17)) should hold. Using $y_{i,1}^{pilot,s} = y_{i,1}^{pilot,s'}$ and $y_{i,1}^{\exp,s} = y_{i,1}^{\exp,s'}$, we can write the following equivalence at $\hat{t} = 1$.

$$\left[\bigwedge_{\tau=1}^{\hat{t}=1} \neg (y_{i,\tau}^{\exp,s} \vee y_{i,\tau}^{pilot,s}) \right] \Leftrightarrow \left[\bigwedge_{\tau=1}^{\hat{t}=1} \neg (y_{i,\tau}^{\exp,s'} \vee y_{i,\tau}^{pilot,s'}) \right] \qquad \text{(A3)}$$

From this equivalence, it follows that $Z_{\hat{t}=1}^{s,s'}$ is true, using (A1), and the conditional non-anticipativity constraints (A4) and (A5), coming from eq. (22), should hold for $\hat{t} = 2$.

$$y_{i,\hat{t}=2}^{pilot,s} = y_{i,\hat{t}=2}^{pilot,s'} \qquad \text{(A4)}$$

$$y_{i,\hat{t}=2}^{\exp,s} = y_{i,\hat{t}=2}^{\exp,s'} \qquad \text{(A5)}$$

Iteratively, it can be shown that these equalities will hold for each $\hat{t} \le t$. This proves proposition 1.

Proposition 2:

$$\left(\underline{\bigvee}_{\tau=1}^{t}(y_{i,\tau}^{oper,s}\,\underline{\vee}\,y_{i,\tau}^{pilot,s})\right)\Rightarrow\left(\underline{\bigvee}_{\tau=1}^{t}(y_{i,\tau}^{oper,s'}\,\underline{\vee}\,y_{i,\tau}^{pilot,s'})\right)\quad \textit{where } s<s' \text{ and } (i,s,s')\in M(i,s,s') \qquad \text{(A6)}$$

Proof:

Let us assume $\left(\underline{\bigvee}_{\tau=1}^{t}(y_{i,\tau}^{oper,s'}\,\underline{\vee}\,y_{i,\tau}^{pilot,s'})\right)$ is true. Then, there has been a pilot plant installation or operation for only one time period for process i until t. Using proposition 1, the scenarios s and s' will be indistinguishable up to $\hat{t}<t$, if there has not been any expansion or pilot plant installation until $\hat{t}$. At $\tau=\hat{t}+1$, following expression must hold because of the conditional nonanticipativity constraints (eq. (22)).

$$\left(\underline{\bigvee}_{\tau=1}^{\hat{t}+1}(y_{i,\tau}^{oper,s'}\,\underline{\vee}\,y_{i,\tau}^{pilot,s'})\right)\Leftrightarrow\left(\underline{\bigvee}_{\tau=1}^{\hat{t}+1}(y_{i,\tau}^{oper,s'}\,\underline{\vee}\,y_{i,\tau}^{pilot,s'})\right) \qquad \text{(A7)}$$

If $(i,s,s')\in M(i,s,s')$ then scenario pairs s and s' are going to be indistinguishable according to (A1) even though they have operated, for one time period. This enforces conditional non-anticipativity constraints (eq. (22)) to hold. Iteratively, this continues until $\hat{t}=t$. This proves proposition 2.

By using the results from proposition 1 and 2, the original logical equation (A1) is reduced to Equation (21). Also, we should note that in proposition 1, variables $y_{i,t}^{pilot,s}$ and $y_{i,t}^{\exp,s}$ are used whereas in proposition 2, $y_{i,t}^{pilot,s}$ and $y_{i,t}^{oper,s}$ has been used. From constraint (19), it is clear that a process must operate if it is expanded, so we could have used $y_{i,t}^{oper,s}$ instead of $y_{i,t}^{\exp,s}$ in proposition 1 as well.

APPENDIX B

Expression (B1) represents the objective of maximizing the expected net present value.

$$\sum_{s \in S} p^s y^{NPV,s} \tag{B1}$$

Expression (B2) calculates the net present value in scenario s.

$$\begin{aligned} y^{NPV,s} = &-\sum_{t=1}^{10}\sum_{i=1}^{3}\left(FE_{i,t}\, y_{i,t}^{\exp,s} + VE_{i,t}\, y_{i,t}^{QE,s}\right) - \sum_{t=1}^{10}\sum_{i=1}^{3} FO_{i,t}\, y_{i,t}^{oper,s} \\ &-\sum_{t=1}^{10}\sum_{k=1}^{8} VO_{j,t}^{y}\, y_{k,t}^{rate,s} - \sum_{t=1}^{10}\left(\alpha_t x_t^{purch,s} - \beta_t x_t^{sales,s} + \gamma_t y_t^{inv,s}\right) \end{aligned} \qquad \forall s \in S \tag{B2}$$

Equations (B3)-(B5) represent the yield relationships for the three processes at each time period and scenario.

$$y_{3,t}^{rate,s} = \theta_{1,step}^{s}\; y_{1,t}^{rate,s} \qquad \forall s \in S, \forall t \in T \tag{B3}$$

$$y_{4,t}^{rate,s} = \theta_{2,step}^{s}\; y_{2,t}^{rate,s} \qquad \forall s \in S, \forall t \in T \tag{B4}$$

$$y_{8,t}^{rate,s} = \theta_{3,step}^{s}\; y_{7,t}^{rate,s} \qquad \forall s \in S, \forall t \in T \tag{B5}$$

Equations (B6)-(B7) are the mass balance constraints at node1 and node2 respectively.

$$y_{5,t}^{rate,s} = y_{3,t}^{rate,s} + y_{4,t}^{rate,s} \qquad \forall s \in S, \forall t \in T \tag{B6}$$

$$y_{7,t}^{rate,s} = y_{5,t}^{rate,s} + y_{6,t}^{rate,s} \qquad \forall s \in S, \forall t \in T \tag{B7}$$

Equation (B8) is the balance constraint that relates inventory, sales and production at consecutive time periods in the network.

$$y_t^{inv,s} = y_{t-1}^{inv,s} + y_{8,t}^{rate,s} + x_t^{purch,s} - x_t^{sales,s} \qquad \forall s \in S, \forall t \in T \tag{B8}$$

Constraint (B9) forces the sales to satisfy demand exactly for each time period and scenario.

$$x_t^{sales,s} = d_t \qquad \forall s \in S, \forall t \in T \tag{B9}$$

In the constraints (B10)-(B12), the outflow from each process is restricted by the capacity of the corresponding process.

$$y_{3,t}^{rate,s} \leq y_{1,t}^{cap,s} \qquad \forall s \in S, \forall t \in T \qquad \text{(B10)}$$

$$y_{4,t}^{rate,s} \leq y_{2,t}^{cap,s} \qquad \forall s \in S, \forall t \in T \qquad \text{(B11)}$$

$$y_{8,t}^{rate,s} \leq y_{3,t}^{cap,s} \qquad \forall s \in S, \forall t \in T \qquad \text{(B12)}$$

The capacities of all the processes in time period t are computed by the equality constraint (B13).

$$y_{i,t}^{cap,s} = y_{i,t-1}^{cap,s} + y_{i,t}^{QE,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad \text{(B13)}$$

Upper and lower bounds for capacity expansions are calculated by constraint (B14).

$$L_i^{QE} y_{i,t}^{\exp,s} \leq y_{i,t}^{QE,s} \leq U_i^{QE} y_{i,t}^{\exp,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad \text{(B14)}$$

Constraints (B15)-(B17) represent the upper and lower bounds for outflows when the process is operating.

$$L_1^{\inf low} y_{1,t}^{oper,s} \leq y_{3,t}^{rate,s} \leq U_1^{\inf low} y_{1,t}^{oper,s} \qquad \forall s \in S, \forall t \in T \qquad \text{(B15)}$$

$$L_2^{\inf low} y_{2,t}^{oper,s} \leq y_{4,t}^{rate,s} \leq U_2^{\inf low} y_{2,t}^{oper,s} \qquad \forall s \in S, \forall t \in T \qquad \text{(B16)}$$

$$L_3^{\inf low} y_{3,t}^{oper,s} \leq y_{8,t}^{rate,s} \leq U_3^{\inf low} y_{3,t}^{oper,s} \qquad \forall s \in S, \forall t \in T \qquad \text{(B17)}$$

Logical constraint (B18) states that operating a process at time period t requires an expansion for any period $\tau = 1,...,t$.

$$y_{i,t}^{oper,s} \Rightarrow \bigvee_{\tau=1}^{t} y_{i,\tau}^{\exp,s} \qquad \forall i \in IU, \forall s \in S, \forall t \in T \qquad \text{(B18)}$$

Constraint (B19) forces a process to be operated in time period t if its capacity is expanded in that time period.

$$y_{i,t}^{\exp,s} \Rightarrow y_{i,t}^{oper,s} \qquad \forall i \in I, \forall s \in S, \forall t \in T \qquad \text{(B19)}$$

Constraints (B20)-(B24) are non-anticipativity constraints for first period decisions.

$$y_{i,1}^{oper,s} = y_{i,1}^{oper,s'} \qquad \forall i \in I, \forall (s,s'), s < s' \qquad \text{(B20)}$$

$$y_{i,1}^{pilot,s} = y_{i,1}^{pilot,s'} \qquad \forall i \in IU, \forall (s,s'), s < s' \qquad \text{(B21)}$$

$$y_{i,1}^{\exp,s} = y_{i,1}^{\exp,s'} \qquad \forall i \in I, \forall (s,s'), s < s' \qquad \text{(B22)}$$

$$y_{i,1}^{QE,s} = y_{i,1}^{QE,s'} \qquad \forall i \in I, \forall (s,s'), s < s' \qquad \text{(B23)}$$

$$y_{k,1}^{rate,s} = y_{k,1}^{rate,s'} \qquad \forall k \in DK, \forall (s,s'), s < s' \qquad \text{(B24)}$$

Constraint (B25) states that at most one pilot plant can be installed for each process having uncertain yield, for each scenario through the entire project life.

$$\sum_{\tau=1}^{10} y_{i,\tau}^{pilot,s} \le 1 \qquad \forall i \in IU, \forall s \in S \qquad \text{(B25)}$$

Constraint (B26) shows that if there has been an expansion in the process until time period t, then there is no need for a pilot plant for a process having uncertain yield at time period t, scenario s.

$$\bigvee_{\tau=1}^{t} y_{i,\tau}^{\exp,s} \Rightarrow \neg y_{i,t}^{pilot,s} \qquad \forall i \in IU, \forall s \in S, \forall t \in T \qquad \text{(B26)}$$

Logic constraint (B27) states that for a given process having uncertain yield, if there is a pilot plant installation at time period t, any expansion cannot be made in the same period.

$$y_{i,t}^{pilot,s} \Rightarrow \neg y_{i,t}^{\exp,s} \qquad \forall i \in IU, \forall s \in S, \forall t \in T \qquad \text{(B27)}$$

Logic constraint (B28) states that, s and s' can be indistinguishable in two ways. The first one is that there is no expansion or pilot plant installation in a process that differentiates s and s'. The second way is that process i, that differentiates the scenarios s and s', is operated for just one period or a pilot plant for process i is installed, and s and s' are two scenarios such that they cannot be differentiated in the second step of uncertainty reduction. Other than these two, scenarios s and s' will be distinguishable.

$$Z_t^{s,s'} \Leftrightarrow \bigwedge_{i \in D(s,s')}^{I} \left\{ \left[\bigwedge_{\tau=1}^{t} \neg (y_{i,\tau}^{\exp,s} \vee y_{i,\tau}^{pilot,s}) \right] \underline{\vee} \left[\left(\underline{\bigvee_{\tau=1}^{t}} (y_{i,\tau}^{oper,s} \underline{\vee} y_{i,\tau}^{pilot,s}) \right) \wedge \left((i,s,s') \in M(i,s,s') \right) \right] \right\}$$

$$\forall (s,s'), s < s', \forall t \in T \qquad \text{(B28)}$$

Disjunctive constraint (B29) shows the decisions are the same for time period t and t+1 if s and s' are indistinguishable at time period t.

$$\begin{bmatrix} Z_t^{s,s'} \\ x_t^{purch,s} = x_t^{purch,s'} \\ x_t^{sales,s} = x_t^{sales,s'} \\ y_{i,t+1}^{oper,s} = y_{i,t+1}^{oper,s'} \quad if\ t \le T-1, \forall i \in I \\ y_{i,t+1}^{\exp,s} = y_{i,t+1}^{\exp,s'} \quad if\ t \le T-1, \forall i \in I \\ y_{i,t+1}^{pilot,s} = y_{i,t+1}^{pilot,s'} \quad if\ t \le T-1, \forall i \in IU \\ y_{i,t+1}^{QE,s} = y_{i,t+1}^{QE,s'} \quad if\ t \le T-1, \forall i \in I \\ y_{k,t+1}^{rate,s} = y_{k,t+1}^{rate,s'} \quad if\ t \le T-1, \forall k \in DK \end{bmatrix} \vee \left[\neg Z_t^{s,s'}\right] \quad \forall (s,s'), s < s', \forall t \in T \qquad \text{(B29)}$$

APPENDIX C

Conversion of logic constraint (21) in the model into binary representation:

$$Z_t^{s,s'} \Leftrightarrow \bigwedge_{i\in D(s,s')}^{I} \left\{ \underbrace{\left[\bigwedge_{\tau=1}^{t} \neg\left(y_{i,\tau}^{\exp,s} \vee y_{i,\tau}^{pilot,s}\right)\right]}_{x_{1,i,t}^{s}} \underline{\vee} \underbrace{\left[\underbrace{\left(\underline{\bigvee}_{\tau=1}^{t} \left(y_{i,\tau}^{oper,s} \underline{\vee} y_{i,\tau}^{pilot,s}\right)\right)}_{x_{3,i,t}^{s}} \wedge \underbrace{\left((i,s,s') \in M(i,s,s')\right)}_{x_{4,i,t}^{s,s'}} \right]}_{x_{2,i,t}^{s,s'}} \right\} \qquad \text{(C1)}$$

If there is no expansion or pilot plant installation then $x_{1,i,t}^{s}$ will be true, if there is any expansion or pilot plant installation then $x_{1,i,t}^{s}$ will be false.

$$\sum_{\tau=1}^{t} \left(y_{i,\tau}^{\exp,s} + y_{i,\tau}^{pilot,s}\right) \le CARD(t)\left(1 - x_{1,i,t}^{s}\right) \qquad \text{(C2)}$$

$$\left(1 - x_{1,i,t}^{s}\right) \le \sum_{\tau=1}^{t} \left(y_{i,\tau}^{\exp,s} + y_{i,\tau}^{pilot,s}\right) \qquad \text{(C3)}$$

If the process is operated for just one period or only the pilot plant has been installed then $x_{3,i,t}^{s}$ will be true, otherwise it will be false. $y_{1,i,t}^{s}$ and $y_{2,i,t}^{s}$ are binary variables used to represent that the sum can be either 0 or greater than 2 when $x_{3,i,t}^{s}$ is 0.

$$\sum_{\tau=1}^{t} \left(y_{i,\tau}^{oper,s} + y_{i,\tau}^{pilot,s}\right) \le 1 + CARD(t)\left(1 - x_{3,i,t}^{s}\right) \qquad \text{(C4)}$$

$$\sum_{\tau=1}^{t} \left(y_{i,\tau}^{oper,s} + y_{i,\tau}^{pilot,s}\right) \ge 1 - CARD(t)\left(1 - x_{3,i,t}^{s}\right) \qquad \text{(C5)}$$

$$\sum_{\tau=1}^{t} \left(y_{i,\tau}^{oper,s} + y_{i,\tau}^{pilot,s}\right) \ge 2 y_{1,i,t}^{s} \qquad \text{(C6)}$$

$$\sum_{\tau=1}^{t} \left(y_{i,\tau}^{oper,s} + y_{i,\tau}^{pilot,s}\right) \le CARD(t)\left(1 - y_{2,i,t}^{s}\right) \qquad \text{(C7)}$$

$$y_{1,i,t}^{s} + y_{2,i,t}^{s} + x_{3,i,t}^{s} = 1 \qquad \text{(C8)}$$

$x_{4,i,t}^{s,s'}$ is equal to 1 if s and s' are two scenario pairs are indistinguishable when process i is either operated for just one period or a pilot plant is installed.

$$x_{4,i,t}^{s,s'} = 1 \quad if \ (i,s,s') \in M(i,s,s') \tag{C9}$$

If $x_{3,i,t}^{s}$ and $x_{4,i,t}^{s,s'}$ are both true then, $x_{2,i,t}^{s,s'}$ will be true. If either $x_{3,i,t}^{s}$ or $x_{4,i,t}^{s,s'}$ is false then $x_{2,i,t}^{s,s'}$ will be false.

$$(1 + x_{2,i,t}^{s,s'}) \geq x_{3,i,t}^{s} + x_{4,i,t}^{s,s'} \tag{C10}$$

$$x_{3,i,t}^{s} \geq x_{2,i,t}^{s,s'} \tag{C11}$$

$$x_{4,i,t}^{s,s'} \geq x_{2,i,t}^{s,s'} \tag{C12}$$

At most one of the possible stages should hold: either there is no expansion or, there is one expansion or operation.

$$x_{1,i,t}^{s} + x_{2,i,t}^{s,s'} \leq 1 \tag{C13}$$

If we represent the right hand side by $\hat{z}_{i,t}^{s,s'}$,

$$\left\{ \left[\bigwedge_{\tau=1}^{t} \neg (y_{i,\tau}^{\exp,s} \vee y_{i,\tau}^{pilot,s}) \right] \underline{\vee} \left[\left(\underline{\bigvee_{\tau=1}^{t}} (y_{i,\tau}^{oper,s} \underline{\vee} y_{i,\tau}^{pilot,s}) \right) \wedge \left((s,s') \in M(i,s,s') \right) \right] \right\} \equiv \hat{z}_{i,t}^{s,s'} \tag{C14}$$

We can write the expression (21) as,

$$Z_{t}^{s,s'} \Leftrightarrow \bigwedge_{i \in D(s,s')}^{I} \hat{z}_{i,t}^{s,s'} \tag{C15}$$

If $x_{1,i,t}^{s}$ or $x_{2,i,t}^{s,s'}$ are true then, $\hat{z}_{i,t}^{s,s'}$ will be true. If either $x_{1,i,t}^{s}$ or $x_{2,i,t}^{s,s'}$ is false, then $\hat{z}_{i,t}^{s,s'}$ will be false. This is captured by equations (C16) – (C18).

$$x_{1,i,t}^{s} \leq \hat{z}_{i,t}^{s,s'} \tag{C16}$$

$$x_{2,i,t}^{s,s'} \leq \hat{z}_{i,t}^{s,s'} \tag{C17}$$

$$x_{1,i,t}^{s} + x_{2,i,t}^{s,s'} \geq \hat{z}_{i,t}^{s,s'} \tag{C18}$$

If for all the processes that distinguish s and s', $\hat{z}_{i,t}^{s,s'}$ is true then $Z_t^{s,s'}$ will be true, otherwise it will be false. This is captured by equations (C19) – (C20).

$$\hat{z}_{i,t}^{s,s'} \geq Z_t^{s,s'} \tag{C19}$$

$$\sum_{i \in D(s,s')} (1 - \hat{z}_{i,t}^{s,s'}) + Z_t^{s,s'} \geq 1 \tag{C20}$$