CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Two Heuristic Algorithms for a Large-Scale Mixed-Integer Production Planning Model with Random Yield and Demand: A Case in Sawmills

Masoumeh Kazemi Zanjani
Mustapha Nourelfath
Daoud Aït-Kadi

May 2010

CIRRELT-2010-23

Bureaux de Montréal :
Université de Montréal
C.P. 6128, succ. Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Two Heuristic Algorithms for a Large-Scale Mixed-Integer Production Planning Model with Random Yield and Demand: A Case in Sawmills

Masoumeh Kazemi Zanjani*, Mustapha Nourelfath, Daoud Aït-Kadi

Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Mechanical Engineering, 1065, avenue de la Médecine, Université Laval, Québec (Québec) G1V 0A6

Abstract. This study considers a real-world multi-period, multi-product production planning problem involving set-up constraints, with random yield and demand. The resulting large-scale multi-stage stochastic mixed-integer model cannot be solved by using the mixed-integer solver of a commercial optimization package. The production planning model is a mixed-integer programming (MIP) model without any special structure. As a consequence, developing efficient decomposition and cutting plane algorithms to obtain a good solution in a reasonable amount of time is not straightforward. We use two solution strategies to find good solutions with an acceptable gap to the optimal solution: (1) The first strategy is based on the progressive hedging algorithm (PHA). The solution of this strategy is a local optimum and an upper bound for the optimal objective value of the multi-stage stochastic model. (2) The second strategy is a successive approximation heuristic which solves the problem by considering only a subset of scenarios which is updated at each iteration. We introduce scenario selection rules in order to increase the rate of convergence and the quality of solution. Computational experiments for a real world large-scale sawmill production planning model verify and compare the effectiveness of the two proposed solution strategies in finding quickly good approximate solutions.

Keywords. Production planning, random yield, sawmill, stochastic programming, sample average approximation, Monte Carlo simulation.

Acknowledgements. This work was supported by Forest E-business Research Consortium (FOR@C) of Université Laval.

---

* Corresponding author: Masoumeh.KazemiZanjani@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec,
Bibliothèque et Archives Canada, 2010

1- Introduction

Stochastic programming (Birge, 1997) and robust optimization (Mulvey et al., 1995) has been applied in several production planning contexts under uncertain environments (see for example, Escudero et al. (1993); Bakir and Byrne (1998); Huang K. (2005); Brandimarte (2006); Mula et al. (2006); Leung et al. (2007)). These applications and others have demonstrated that incorporating uncertainty into production planning models is significant because it has been demonstrated that in most of real-world applications, stochastic models propose more robust production plans than their deterministic counterparts. However, such stochastic approaches may lead to models of non manageable sizes. In particular, when random parameters are characterized by a dynamic behavior throughout the time, the decision process should take place in multiple stages of time, as the information is revealed. The resulting multi-stage stochastic program becomes computationally intractable as the number of scenarios increases. The mentioned complexity is even intensified when the production planning problem is formulated as a multi-stage stochastic mixed-integer (MS-MIP) program.

In this paper we study a multi-period production planning problem that is motivated by a real life case study, and that results in a large-scale MS-MIP. In this problem, alternative processes produce multiple products from several classes of raw materials. However, regarding the set-up considerations, only one family of processes can be run in each period. In each period in the planning horizon, we are looking for the selection of a family of processes, the number of times each process should be run, as well as the consumption of each material class. The objective is to minimize products inventory/backorder and raw material consumption costs, regarding fulfillment of products demands, machine capacities, and material inventory. We study a manufacturing environment where the characteristics of raw materials are non-homogeneous and random. Thus, the quantities of products that can be produced by each process (process yields) are random variables. Moreover, the demand is also uncertain and non-stationary during the planning horizon. This study is motivated by the production planning problem in sawmills. The raw materials in sawmills are the logs which have non-homogeneous characteristics. Moreover, the lumber demand in the market is also uncertain.

Kazemi Zanjani et al. (2010a,b,c) developed a set of stochastic and robust optimization models to deal with different aspects of production planning in sawmills under uncertainty. In Kazemi Zanjani et al. (2010c), a two-stage stochastic programming model is proposed to deal with the uncertainty in the quality of raw materials, while a robust optimization approach is used in Kazemi Zanjani et al. (2010a) to find a more robust production plan in this problem. In Kazemi Zanjani et al. (2010b), a multi-stage stochastic programming model is proposed considering that both raw materials and demand are random. Unlike Kazemi zanjani et al. (2010a,b,c), we address this problem by taking into account set-up constraints. This is an important extension to widen the application of our approach. In fact, regarding the technological constraints for changing machine set-ups in some sawmills, only one process family including processes with similar set-up requirements can be executed in each period. As a consequence, set-up constraints should be taken into account in the production planning model.

We model the problem as a multi-stage mixed-integer stochastic program with full recourse based on a hybrid scenario tree for the random yield and demand. While the newly developed model is applicable to a larger number of sawmill cases, a challenging problem is faced for solving real-life instances in reasonable time. In this paper, we consider a realistic large-scale problem as the case study. A mixed-integer solver of a commercial optimization package, namely CPLEX MIP solver is not capable to solve the resulting deterministic equivalent model, due to memory problem. Thus, other solution strategies should be considered to find good approximate solutions.

The development of efficient algorithms which are able to solve stochastic mixed-integer programs (MIP) with a large number of scenarios is one of the challenges in stochastic programming. Løkketangen and Woodruff (1996) proposed a heuristic which combines the progressive hedging algorithm (PHA) (proposed in Rockafellar and Wets (1991)) with tabu search in order to solve a multi-stage stochastic MIP with binary variables. Haugen et al. (2001) casted the PHA in a meta-heuristic framework with the sub-problems generated for each scenario solved heuristically. In Lulli and Sen (2004), a branch-and-price algorithm is proposed to solve special structured multi-stage MSMIP problems. Lulli and Sen (2006) proposed a heuristic, referred as the scenario updating (SU) procedure, to solve large stochastic MIP models with complete recourse. For recent surveys on stochastic mixed-integer programs, the reader may refer to Klein Haneveld and Van der Vlerk (1999), Schultz (2003) and Sen (2003).

Most of the existing MIP algorithms, namely, decomposition and cutting plan algorithms (e.g., branch and cut, branch and price, etc.) are more efficient for special structured MIP models (e.g., lot-sizing and batch sizing problems). However, the problem studied in this paper is a general multi-stage stochastic mixed integer model with *no special structure*. Thus, using these sophisticated algorithms for our problem cannot guarantee the convergence to a good solution in reasonable time. In this article, two efficient solution strategies are developed and compared for a real sawmill example.

The first solution strategy is a scenario decomposition approach that decomposes the stochastic model for the possible scenarios of the random events. Each scenario sub-problem then becomes a deterministic problem that can either be solved directly by CPLEX MIP solvers or an efficient heuristic algorithm. Following the original decomposition scheme proposed by Rockafellar and Wets (1991), we address the issue of using globally the local information yielded by the sub-problems to guide the overall algorithm toward a unique solution. By the non-anticipativity condition in multi-stage stochastic programming, the solutions of indistinguishable scenarios at each node of the scenario tree should be identical. Thus, the non-consensus amongst scenario sub-problem solutions in each node of the scenario tree is penalized by adding quadratic terms in the objective function of each scenario sub-problem. This corresponds to an augmented Lagrangean strategy, which is used in the progressive hedging algorithm (PHA) proposed by Rockafellar and Wets (1991) for general multi-stage stochastic programs. As in PHA, our solution strategy ensures implementable solutions at all iterations. By adding terms in the objective function to penalize lack of implementability, the algorithm may come up with a "well hedged" solution that can perform well under all scenarios. It should be mentioned that this solution strategy converges to a global optimum in the continuous convex case. In the non-convex case it converges to a local optimum if all sub-problem are solved for a local optimal. If we solve the scenario sub problems for optimality, the converged solution can be considered as an upper bound for the optimal solution.

The second solution strategy is inspired by the Scenario Updating (SU) method proposed by Lulli and Sen (2006). It is based on solving instances of the problem, which contain only a subset of scenarios in the scenario tree. The subset of scenarios is updated by adding those scenarios, which imply either certain degradation or an improvement of the objective function value. We improve the SU algorithm in the following directions. The first improvement is motivated by the fact that, in our experiments, the scenario selection rules proposed in Lulli and Sen (2006) for updating the scenario tree did not converge fast. We propose a new scenario selection rule and we combine it with some of the most appropriate rules in SU method in order to increase the convergence rate of the algorithm as well as the quality of the approximate solution. Furthermore, the proposed lower bound on the objective function value of a stochastic MIP model in Lulli and Sen (2006) is more appropriate for a simple recourse multi-stage stochastic MIP. As the problem we are addressing is a multi-stage stochastic MIP with full recourse, we consider only the proposed upper bound to evaluate different scenarios to be added into the current scenario tree. The approximate solution of the second solution strategy can be considered as a lower bound for the optimal solution, if an appropriate subset of scenarios is taken into account.

Both solution strategies are applied for sawmill production planning under the uncertainty in raw material (log) quality and product (lumber) demand. Numerical results for a prototype sawmill indicate that the second solution strategy results an approximate solution by considering a small number of scenarios in the scenario tree. Moreover, an acceptable gap is observed between the approximate solution and the upper bound of the optimal solution computed by the first strategy.

The contribution of this paper is then threefold. First, we develop a framework based on the PHA that takes advantages of efficient methods to solve the deterministic version of the problem under study, and it can be implemented on parallel machines. This kind of general framework was proposed in Løkketangen and Woodruff (1996) and was successfully applied to the problem of stochastic network design in Cranic et al. (2009). To the best of our knowledge, this paper is the first to propose such approach for the case of stochastic production planning in sawmills. Second, we extend the SU method to increase its convergence rate in the context of full recourse stochastic models. Third, we apply the proposed model and algorithms for a real-life industrial case.

The rest of the paper is organized as follows. In section 2, we provide a multi-stage stochastic mixed-integer program for MPMP production planning with random yield and demand. Sections 3 describe the sawmill case study, while sections 4 and 5 present, respectively, the proposed PHA and the extended SU method. In section 6, the implementation results of the MS-MIP model and the solution methods for a prototype realistic-scale sawmill are presented. Finally, conclusions are given in section 7.

2- General formulation of the problem

In this section, we first present a deterministic mathematical formulation for the problem under consideration. Then, we provide the multi-stage stochastic formulation to address the problem by considering the random yield and demand.

2-1- A deterministic model

Consider a production unit with a set of products P, a set of classes of raw materials C, a set of production processes A, a set of process families F, a set of machines R, and a planning horizon consisting of T periods. To state the deterministic linear programming (LP) model for this problem, the following notations are used:

2-1-1- Notations

<u>*Indices*</u>

p product

t period

c raw material class

a production process

r machine

f family of process

f_a family of process a

<u>*Parameters*</u>

h_{pt} inventory holding cost per unit of product p in period t

b_{pt} backorder cost per unit of product p in period t

m_{ct} raw material cost per unit of class c in period t

I_{c0} the inventory of raw material class c at the beginning of planning horizon

I_{p0} the inventory of product p at the beginning of planning horizon

s_{ct} the quantity of material of class c supplied at the beginning of period t

d_{pt} demand of product p by the end of period t

ϕ_{ac} the units of class c raw material consumed by process a

ρ_{ap} the units of product p produced by process a (yield of process a)

δ_{ar} the capacity consumption of machine r by process a

M_{rt} the capacity of machine r in period t

M a large positive number

Decision variables

X_{at} the number of times each process a should be run in period t (production plan)

$$Y_{ft} = \begin{cases} 1, & \text{if the family } f \text{ of processes is selected in period } t \\ 0, & \text{otherwise} \end{cases}$$

I_{ct} inventory size of raw material of class c by the end of period t

I_{pt} inventory size of product p by the end of period t

B_{pt} backorder size of product p by the end of period t

2.1.2 The deterministic LP model

$$\text{Minimize } Z = \sum_{p \in \mathrm{P}} \sum_{t=1}^{T} (h_{pt} I_{pt} + b_{pt} B_{pt}) + \sum_{c \in \mathrm{C}} \sum_{t=1}^{T} \sum_{a \in A} m_{ct} \phi_{ac} X_{at}, \tag{1}$$

Subject to

$$I_{ct} = I_{ct-1} + s_{ct} - \sum_{a \in A} \phi_{ac} X_{at}, \qquad t = 1,\ldots,T, c \in \mathrm{C}, \tag{2}$$

$$I_{p1} - B_{p1} = I_{p0} + \sum_{a \in \mathrm{A}} \rho_{ap} X_{a1} - d_{p1},$$

$$I_{pt} - B_{pt} = I_{pt-1} - B_{pt-1} + \sum_{a \in \mathrm{A}} \rho_{ap} X_{at} - d_{pt}, \qquad t = 2,\ldots,T,\ p \in \mathrm{P}, \tag{3}$$

$$\sum_{a \in \mathrm{A}} \delta_{ar} X_{at} \le M_{rt}, \qquad t = 1,\ldots,T,\ r \in \mathrm{R}, \tag{4}$$

$$X_{at} \le M Y_{f_a t} \qquad t = 1,\ldots,T,\ a \in \mathrm{A}, \tag{5}$$

$$\sum_{f \in F} Y_{ft} = 1 \qquad t = 1,\ldots,T, \tag{6}$$

$$X_{at} \ge 0, I_{ct} \ge 0, I_{pt} \ge 0, B_{pt} \ge 0, Y_{ft} \in \{0,1\}, \quad t = 1,\ldots,T,\ p \in \mathrm{P}, c \in \mathrm{C},\ a \in A, f \in F. \tag{7}$$

The objective function (1) minimizes the total inventory and backorder costs for all products and raw material cost for all classes in the planning horizon. Constraint (2) corresponds to the raw material inventory balance. Constraint (3) indicates the products inventory balance. Constraint (4) requires that the total production do not exceed the available production capacity. Constraints (5) and (6) require that only one family of processes that require the same set-up can be executed in each period.

2-2- Multi-stage stochastic programming extension

In this section, we first describe a summary of the proposed approach to model the random yield and demand. Then we provide the production planning formulation by multi-stage stochastic programming. For more details, the reader may refer to Kazemi Zanjnai et al. (2010b), where this formulation is used without set-up constraints.

2-2-1- Modeling the random yield and demand

It is assumed that the random demand has a non-stationary behavior during the planning horizon. Thus we model the random demand as a scenario tree. The nodes at stage *t* of the tree constitute the states (scenarios) of demand. The leaf nodes in the scenario tree are called *scenarios*. The planning horizon is clustered into N stages, where each stage includes a number of periods. In other words, it is supposed that the random demand is stationary during the time periods at each stage.

On the other hand, we assume that raw materials are supplied from the same supply source during the planning horizon. Thus, it is supposed that the random yield has a stationary probability distribution. A number of scenarios are taken in to account for yields by discritization of the original probability distribution.

In order to have a single stochastic production planning model that considers the random yield and demand, yield scenarios are integrated with the demand scenario tree, forming a hybrid scenario tree. At each node of the tree, which denotes one demand scenario for the corresponding stage, different yield scenarios can take place. However, regarding the stationary behavior of the random yield, only one of the yield scenarios might be observed during the planning horizon.

2-2-2- Multi-stage stochastic program

We now formulate the problem as a multi-stage stochastic (MSP) model based on the hybrid scenario tree for the random yield and demand. It should be noted that we consider a *full recourse* multi-stage stochastic model with respect to demand scenarios. In other words, it is supposed at the start of the first period of each stage, enough information is available to the decision maker to know which demand scenario is in force for that stage. So the decision maker can adjust the production plan X_{at} for different demand scenarios. A decision X_{at} is said to be "implementable" if it cannot distinguish between different scenarios that at a particular stage of time are regarded as indistinguishable from each other on the basis of information so far available. There are two approaches to impose the non-anticipativity condition in the multi-stage stochastic programs which lead to *split variable* formulation and *compact* formulation. In this section, a compact formulation is used to represent the problem and the decision variables X_{at} and Y_{ft} are then defined for each node of the demand scenario tree.

On the other hand, the model is *simple recourse* with respect to yield scenarios. In fact, as the quality of materials is not is not known before being processed, the yield scenarios can only be revealed after implementation of the production plan. Thus, the production plan for each node of the demand scenario tree should be fixed for yield scenarios. It is evident that the inventory and backorder sizes of products in each period, which are the state variables, depend on the demand scenarios as well as yield scenarios, thus they are indexed for yield scenarios as well as demand nodes. Regarding the above discussions, the following notations in addition to those provided in subsection 2.1.1 are used in the multi-stage stochastic model. The compact formulation of the multi-stage stochastic model follows these notations.

2-2-2-1- Notations

Indices

Tree scenario tree.

S number of scenarios for the random yields.

i scenario of the random yield.

n,m node of the scenario tree.

$a(n)$ immediate predecessor of node *n* in the scenario tree.

t_n set of time periods corresponding to node *n* in the scenario tree.

Parameters

$d_{pt}(n)$ demand of product *p* by the end of period *t* at node *n* of the scenario tree.

$p(n)$ probability of node *n* of the scenario tree.

p^i probability of scenario *i* for the random yield.

Decision variables

$X_{at}(n)$ the number of times each process a should be run in period t at node n of the scenario tree.

$$Y_{ft}(n) = \begin{cases} 1, & \text{if the family } f \text{ of processes is selected at node } n \text{ in period } t \\ 0, & \text{otherwise} \end{cases}$$

$I_{ct}(n)$ inventory size of raw material of class c by the end of period t at node n of the scenario tree.

$I^{i}_{pt}(n)$ inventory size of product p by the end of period t for scenario i of the random yield at node n of the scenario tree.

$B^{i}_{pt}(n)$ backorder size of product p by the end of period t for scenario i of the random yield at node n of the scenario tree.

2-2-2-2- Multi-stage stochastic model (compact formulation)

$$\text{Minimize } Z = \sum_{n \in Tree} p(n)(\sum_{t \in t_n} \sum_{c \in \mathrm{C}} \sum_{a \in A} m_{ct} \phi_{ac} X_{at}(n)) + \sum_{n \in Tree} p(n)(\sum_{i=1}^{S} p^{i} (\sum_{t \in t_n} \sum_{p \in \mathrm{P}} (h_{pt} I^{i}_{pt}(n) + b_{pt} B^{i}_{pt}(n)))) \tag{8}$$

Subject to

$$I_{ct}(n) = I_{ct-1}(m) + s_{ct} - \sum_{a \in A} \phi_{ac} X_{at}(n), \quad n \in Tree, t \in t_n, c \in \mathrm{C}, \quad m = \begin{cases} a(n), & t-1 \notin t_n, \\ n, & t-1 \in t_n, \end{cases} \tag{9}$$

$$\sum_{a \in \mathrm{A}} \delta_{ar} X_{at}(n) \le M_{rt}, \qquad n \in Tree, t \in t_n, r \in \mathrm{R}, \tag{10}$$

$$I^i_{pt}(n) - B^i_{pt}(n) = I^i_{pt-1}(m) - B^i_{pt-1}(m) + \sum_{a \in \mathrm{A}} \rho^i_{ap} X_{at}(n) - d_{pt}(n),$$
$$n \in Tree, t \in t_n, p \in \mathrm{P}, i = 1,...,S,$$
$$m = \begin{cases} a(n), & t-1 \in t_n, \\ n, & t-1 \notin t_n, \end{cases} \quad (11)$$

$$X_{at}(n) \le MY_{f_a t}(n) \qquad t \in t_n, a \in \mathrm{A}, n \in Tree, \quad (12)$$

$$\sum_{f \in F} Y_{ft}(n) = 1 \qquad t \in t_n, n \in Tree, \quad (13)$$

$$X_{at}(n) \ge 0, Y_{ft}(n) \in \{0,1\}, I_{ct}(n) \ge 0, I^i_{pt}(n) \ge 0, B^i_{pt}(n) \ge 0,$$
$$n \in Tree, f \in F, t \in t_n, c \in C, p \in P, a \in A, i = 1,...,S. \quad (14)$$

The objective function (8) minimizes the expected material cost for demand nodes of the scenario tree and the expected inventory and backorder costs for demand nodes and yield scenarios. In model (8)-(14), the decision variables are indexed for each node, as well as for each time period, since the stages do not correspond to time periods. As it was mentioned in 2.2.1, each node at a stage includes a set of periods which is denoted by t_n. In this model, there are coupling variables between different stages and these are the ending inventory and backorder variables at the end of each stage. As it can be observed in this model, two different node indices (*n* and *m*) are used for inventory/backorder variables in the inventory balance constraints ((9) and (11)). More precisely, for the first period at each stage, the inventory or backorder is computed by considering the inventory or backorder of the last period corresponding to its immediate predecessor node, while for the rest of the periods at that stage, the inventory/backorder size of previous period corresponding to the same node are taken into account.

3- Case study

We consider the sawmill production planning problem. In sawmills, logs are classified according to some attributes namely: diameter class, species, length, taper, etc. Logs are broken down into different dimensions of lumbers by means of different cutting patterns. From each log, several pieces of sawn lumbers are produced depending on the cutting pattern. The lumber quality (grade) as well as its quantity yielded by each cutting pattern depends on the quality and characteristics of input logs. Despite the classification of logs in sawmills, variety of characteristics (in terms of diameter, number of knots, internal defects, etc.) might be observed in different logs of the same class. As it is not possible in many sawmills to scan the logs before planning, the exact yields of cutting patterns for different log classes cannot be determined in priori. Uncertainty of the lumber demand in the market is another important parameter that should be taken into account in sawmill production planning.

The prototype sawmill considered as the case study is a medium capacity softwood sawmill located in Quebec (Canada). It is assumed that the input bucked logs into the sawing unit are categorized into 3 classes. 5 different cutting patterns are available. The production processes belong to 2 distinct set-up families. The sawing unit produces 27 products. We consider two bottleneck machines: Trimmer and Bull. The planning horizon consists of 30 periods (days). The hybrid scenario tree for the random demand and yield in this example is generated as follows.

Demand Scenario tree

We used the existing historical data as well as the insights of the experts in the lumber market to estimate the demand scenario tree. At each stage of the scenario tree, except stage 1, we estimate a normal distribution for products demands (per product, per day). The normal distribution is then approximated by a 3 point discrete distribution (high, average, and low demand) by using the Gaussian quadrature method (Miller and Rice, 1983). We supposed that the demand for the next

10 days has a stationary behaviour, which is a realistic assumption in the lumber market. Thus, we clustered the 30 periods planning horizon into 3 stages and hence 4-stage scenario tree is resulted. The first stage consists of time period zero (present time), the second-stage includes periods 1-10, etc. Moreover, based on the experts' insight, we suppose that the demands for all products are perfectly correlated and all products have the same behaviour at each stage of the scenario tree, while each product has its own daily demand (normal) distribution. In other words, if at stage t, the market condition is good, the demand scenario for all products can be expected to be high. The mentioned approximations results a 4-stage scenario tree including 27 demand scenarios and 40 nodes, as shown in figure 1.

Yield Scenarios

At each node of the demand scenario tree a number of yield scenarios are taken into account. Based on the historical data in Quebec sawmills for the processes yields, normal distributions were estimated for the random yields. The normal distribution corresponding to the yield of each process was then approximated by discrete scenarios. We consider 27 yield scenarios at each node of the demand scenario tree.

The above scenario generation approach for the random demand and yield in this example results a hybrid scenario tree with 40 nodes, where each node includes 3 branches as demand scenarios and 27 branches as yield scenarios. The compact multi-stage stochastic model (8)-(14) for this sawmill example is a mixed-integer program (MIP) with nearly 600,000 decision variables and 300,000 constraints, which cannot be solved by CPLEX 11 due to the lack of memory. In the next section, two efficient heuristics are developed and used in this case study. The first heuristic is a scenario decomposition strategy (section 4), while the second one is a scenario updating strategy (section 5).

Insert figure 1 here.

4- Applying the scenario decomposition method to sawmill production planning

4-1- The general algorithm

The algorithm is based on the principle of scenario decomposition, as initially proposed in the progressive hedging algorithm (PHA) of Rockafellar and Wets (1991) for multi-stage stochastic linear programs. For each scenario, a deterministic optimization sub-problem is generated and solved. We denote by Z(s) the objective function of scenario *s* sub-problem. However, the solution for any particular scenario may not be of any value to us. Since we are not prescient, we have to require solutions that do not necessitate foreknowledge and that will be feasible no matter which scenario is realized. In other words, the solutions should be implementable for all scenarios. The implementability condition indicates that for any pair of scenarios that are indistinguishable at stage *t* (i.e. scenarios which share the same node at stage *t* in the scenario tree), the decision variables corresponding to those scenarios and periods at stage *t* should be identical. The PHA forces implementable solutions at all iterations by adding terms in the objective function to penalize lack of implementability. At each iteration, an estimation of the implementable solution is calculated as the *mean* of the solution values of the scenario sub-problems. The latter are then solved again with adjusted penalties on the difference between the local solution and the average (global) estimation. The general form of the PHA can be represented by the following steps:

<u>Step 1</u> (initialization)

Set the progressive hedging (PH) iteration counter to 0 ($k \leftarrow 0$). Set $w^0(s) \leftarrow 0$.

<u>Step 2</u>

For each scenario *s*, a deterministic optimization sub-problem is solved. The obtained approximate solutions are determined by $X^0(s)$.

Step 3

Increment the PH iteration counter ($k \leftarrow k+1$).

For each node *n* in the scenario tree and for its time index $t = t_n$, evaluate the value of an implementable solution $\bar{X}^{k-1}(t_n, n)$ as the *mean* of all the solution values of scenarios, in the bundle of scenarios that are indistinguishable from *s* at node *n*. For all indistinguishable scenarios at node *n*, set $\bar{X}^{k-1}(s) = \bar{X}^{k-1}(t_n, n)$.

Step 4

Evaluate $w^k(s) \leftarrow w^{k-1}(s) + r(X^{k-1}(s) - \bar{X}^{k-1}(s))$ with $r > 0$.

These values correspond to the dual prices associated with the implementability constraints. In other words, they can be interpreted as the value of information.

Step 5

For each scenario *s*, approximate solutions are obtained for the problem after adding the terms that penalize the lack of implementability in the objective function $Z(s)$. The penalty term is computed as follows:

$$w^k(s)X(s) + \frac{r}{2}\left\|X(s) - \bar{X}^{k-1}(s)\right\|^2, \qquad r \geq 0.$$

The penalty terms correspond to an augmented Lagrangean strategy.

Step 6

If the termination criteria are not met, then go to step 3. The termination criteria are based on convergence. Nevertheless, we can also terminate based on time, since non-convergence is a possibility (Rockafellae and Wets, 1991). Iterations are counted until the counter *k* reaches a predetermined value or the algorithm has converged.

4-2- Application to sawmill production planning

The statement of the above PHA gives rise to two types of optimization sub-problems. The first is a linear MIP problem which is only used at iteration 0 (step 2). The second type which is used in other iterations (step 5) is the modified form of the first type by adding the quadratic penalty terms in the objective function. Before formulating these sub-problems for the sawmill production planning problem, we introduce the required additional notations. To enforce implementability, we act on two decision variables: X_{at} and Y_{ft} . We use w_{at}^{X} and w_{at}^{Y} as the corresponding dual variables corresponding to X_{at} and Y_{ft} , respectively. At each iteration, dual variables are evaluated as described in step 4 of the PHA. Furthermore, in each scenario sub-problem, the decision variables in model (8)-(14) in subsection 2.2 are defined for each demand scenario *s* instead of each node *n* in the scenario tree. The demand parameter d_{pt} is also defined for each scenario.

We are now ready to formulate the scenario sub-problem in step 5 of the PHA. Our hybrid scenario tree includes 4 stages, 27 demand scenarios and 40 nodes (see figure 1). The planning horizon is clustered into 3 stages, where each stage includes 10 periods. In the mentioned scenario tree stage 2 includes periods 1-10 and nodes 2-4, stage 3 includes periods 11-20 and nodes 5-13, etc. The total number of yield scenarios ate each node of the demand scenario is equal to S=27. It is evident that the non-anticipativity constraints should be considered for decision variables corresponding to periods in stages 2 and 3 (periods 1 to 20). The scenario sub-problem in the PHA (except for iteration 0) can be formulated as follows.

Scenario sub-problem in the PHA (for scenario s)

$$\text{Minimize}\, Z(s) = \sum_{t=1}^{T}\sum_{c\in \mathrm{C}}\sum_{a\in \mathrm{A}} m_{ct}\phi_{ac}X_{at}(s) + \sum_{i=1}^{S} p^{i}\sum_{t=1}^{T}\sum_{p\in \mathrm{P}}(h_{pt}I_{pt}^{i}(s) + b_{pt}B_{pt}^{i}(s)) +$$

$$\sum_{a\in A}\sum_{t=1}^{20} w_{at}^{X}(s)X_{at}(s)+\sum_{f\in F}\sum_{t=1}^{20} w_{ft}^{Y}(s)Y_{ft}(s)+ \tag{15}$$

$$(r/2)\sum_{a\in A}\sum_{t=1}^{20}\theta_{at}^{2}(s)+(r/2)\sum_{f\in F}\sum_{t=1}^{20}\gamma_{ft}^{2}(s)$$

Subject to

$$I_{ct}(s)=I_{ct-1}(s)+s_{ct}-\sum_{a\in A}\phi_{ac}X_{at}(s), \qquad t=1,\ldots,T\ , c\in \mathrm{C}, \tag{16}$$

$$\sum_{a\in \mathrm{A}}\delta_{ar}X_{at}(s)\le M_{rt}, \qquad t=1,\ldots,T\ , r\in \mathrm{R}, \tag{17}$$

$$I_{pt}^{i}(s)-B_{pt}^{i}(s)=I_{pt-1}^{i}(s)-B_{pt-1}^{i}(s)+\sum_{a\in \mathrm{A}}\rho_{ap}^{i}X_{at}(s)-d_{pt}(s), \qquad t\in 1,\ldots,T\ , p\in \mathrm{P},\ i=1,\ldots,S, \tag{18}$$

$$X_{at}(s)\le MY_{f_a t}(s) \qquad t=1,\ldots,T\ , a\in \mathrm{A}, \tag{19}$$

$$\sum_{f\in F}Y_{ft}(s)=1 \qquad t=1,\ldots,T\ , \tag{20}$$

$$\theta_{at}(s)=X_{at}(s)-\bar{X}_{at}(s), \qquad t=1,\ldots,20, a\in A, \tag{21}$$

$$\gamma_{ft}(s)=Y_{ft}(s)-\bar{Y}_{ft}(s), \qquad t=1,\ldots,20, f\in F, \tag{22}$$

$$X_{at}(s)\ge 0, Y_{ft}(s)\in\{0,1\}, I_{ct}(s)\ge 0,\ I_{pt}^{i}(s)\ge 0,\ B_{pt}^{i}(s)\ge 0, \qquad t=1,\ldots,T\ , c\in C, p\in P,\ a\in A, f\in F, i=1,\ldots,S, \tag{23}$$

where

$$\bar{X}_{at}(s)=\sum_{s\in B_{s,t}}p^{s}X_{at}(s)/p(A_{s,t}),$$
$$\bar{Y}_{ft}(s)=\sum_{s\in B_{s,t}}p^{s}Y_{ft}(s)/p(A_{s,t}), \tag{24}$$

and

p^s probability of the demand scenario *s* in the scenario tree

$B_{s,t}$ bundle of scenarios that are indistinguishable from *s* in period *t*

$A_{s,t}$ the node in the scenario tree corresponding to scenario *s* in period *t*

$p(A_{s,t})$ the probability of node $A_{s,t}$

As some examples of bundle of indistinguishable scenarios $B_{s,t}$ at stage 2 (periods 1 to 10) in the scenario tree of figure 1, we can refer to scenarios 1 to 9 which share node 2, scenarios 10-18 which share node 3, and scenarios 19-27 which share node 4.

Finally, we need to specify the termination criteria for step 7 of the PHA. In this paper, the PHA is applied for integer variables full convergence. In other words, the convergence criteria is achieved when all integer components of $\bar{X}^k$ are equal to their counterparts in $\bar{X}^{k-1}$. The values of continuous variables are then determined by solving the deterministic equivalent model (8)-(14) when the integer variables are fixed at their converged values.

In this work, the scenario sub problems are solved by CPLEX. However, this solution strategy provides the possibility of using heuristic or meta-heuristic algorithms to solve the scenario sub problems. Moreover, this method can also be implemented on parallel machines which reduces considerably the execution time.

5- Applying the revised scenario updating procedure to sawmill production planning

The scenario updating (SU) procedure proposed by Lulli and Sen (2006) has been motivated by the contamination method (Dupačová, 1995) which has been presented as a tool for post-optimality analysis in scenario-based multi-stage stochastic linear programs (MSLP). The idea behind the method consists of estimating lower and upper bounds for the objective function value of a multi-stage stochastic program when new scenarios are added to the current scenario tree, i.e., it explores the influence of out-of-sample scenarios on the solution of the stochastic program. We describe the SU procedure with the MSLP program given in the following form:

$$\min_{x \in \mathrm{X}} f\ (x\ , P\ ), \tag{25}$$

with x denoting the vector of first-stage decision variables, f linear in P (probability distribution) and $\mathrm{X} \subset R^n$ the set of feasible decision variables. Denoting by $\phi(P)$ the optimal value of the program (25) the contamination method focuses on $\phi(P)$ and its perturbations with respect to inclusion of additional scenarios. Under appropriate assumptions on the optimal solution set and on the value function we can bound the objective function value of the perturbed stochastic program. If we consider the perturbation given by a single scenario $\{z\}$ with probability mass p_z and probability distribution P concentrated at scenarios $\xi^1,...,\xi^P$ with probability mass $p_s\ \forall s \in P$, then the perturbed distribution can be presented as $P' = (1 - p_z)P + p_z z$. Figure 2 illustrates better the perturbation of a current scenario tree P by a scenario $\{z\}$.

Insert figure 2 here.

The lower bound on the objective function value of the perturbed stochastic program is as follows.

$$LB\ = (1 - p_z)\phi(P) + p_z \phi(\mathrm{z}\ ), \tag{26}$$

where $\phi(z)$ is the value of the scenario $\{z\}$ (deterministic) problem. An upper bound on the optimal value of the contaminated program can be computed, if the SMIP has a complete recourse property. We have:

$$UB\ = (1 - p_z)\phi(P) + p_z f\ (x\ (\mathrm{P}), \delta(\mathrm{z}\ )), \tag{27}$$

where $f\ (x\ (P\ ), \delta(z\ ))$ denotes the objective function value of the current solution $x\ (P\ )$ under the perturbed scenario tree with a probability distribution $\delta(z\ )$. The objective function value of current solution in the production planning model (8)-(14) under the perturbed scenario tree is computed as follows:

(1) Formulate model (8)-(14) for the perturbed scenario tree.

(2) Fix the decision variable values corresponding to nodes in the current (non-perturbed) scenario tree, based on the current solution $x(P)$.

(3) Solve model (8)-(14) with decision variables in the nodes corresponding to the added scenario.

For more details on how these bound are derived, the reader is referred to Lulli and Sen (2006). The scenario updating procedure can be summarized as follows.

Initialization

We start with a subset of scenarios $P_0 \subset P$, where P is the set of scenarios (selection phase). Different decision rules can be used for the selection phase.

Step 1

At iteration *k*, solve the “reduced” problem, which is composed by the subset of scenarios P_k , by an appropriate exact/approximate algorithm.

Step 2

Compute an upper and lower bound on the change of the objective value induced by adding a scenario not in the current scenario tree. Those scenarios, which if added to the current scenario tree imply a significant change in the objective function value, are candidates to enter in the next sub-tree (updating phase).

Step 3

If there are no scenario candidates then stop, otherwise add those scenarios or some of them to the current scenario sub-tree, and go to step 1.

It would be worth mentioning that unlike a simple recourse multi-stage stochastic model which considers a fixed first-stage decision variable (e.g., production plan) at the beginning of the planning horizon, a full recourse model updates the decisions as new information is available through the time. Thus, the cost of a multi-stage stochastic model with simple recourse is usually higher than the cost of a full recourse model (it was shown in Kazemi Zanjani et al. (2010b)). Moreover, the cost of the deterministic model for some scenarios can be considerably high, comparing to the cost of a multi-stage stochastic model with full recourse. As a consequent, the lower bound defined in (26) might not be appropriate for a full recourse MS-MIP model. More precisely, this lower bound can be much higher than the objective value of the MS-MIP model with full recourse after adding a new scenario in to the current scenario tree. On the other hand, both terms in the upper bound defined in (27), $\phi(P)$ and $f\,(x\,(\mathrm{P}),\delta(z\,))$, as well as ϕ_{δ_z} (optimal objective function value of perturbed scenario tree) can be computed for a full recourse MS-MIP model. Thus, the proposed upper bound can be considered as a more realistic measure to verify the impact of adding a new scenario in the scenario tree.

Scenario selection rules

The key issue for an efficient implementation of the scenario updating method is concerned with the selection of representative scenarios. The goal is to select those scenarios which provide a good approximation of the scenario tree, in terms of quality of the solution. In the following, several decision rules to handle this task are discussed. It should be noted that in Lulli and Sen (2006) the following scenario selection rules are proposed:

- The *highest probability* rule: Select scenarios in the decreasing order of their probability.

- The *dissimilarity* rule: Solve the deterministic problem for each scenario and compare their first-stage decisions. Select those scenarios which have the most different here and now solutions.
- The *random rule*: select scenarios in a random manner.
- The *mixed* rule: Use an appropriate mix of the other rules.

As it was mentioned before, the production planning model in this article is formulated as a full recourse stochastic model. In other words, we do not have any first-stage decision variable in this model. As a consequent we use the *dissimilarity* rule based on the objective function value of each scenario deterministic model. Moreover, some scenarios with small probabilities might have very large or very small objective values. By integrating them into the scenario tree they might not have considerable impact on the objective function value of the full recourse stochastic model, due to their low probability. Thus, we did not find the *dissimilarity* rule as an efficient rule in a full recourse stochastic model to accelerate the rate of convergence of the algorithm.

In order to increase the convergence rate of the SU algorithm and also to improve the quality of solutions we propose a new scenario selection rule that we refer it as the *stage priority* rule and is described as follows.

- The *stage priority* rule: We define the priority of stages in the scenario tree in their increasing order as follows: Stage 1 has the highest priority, while the last stage has the lowest one. Thus, scenarios should be selected so that the majority of nodes corresponding to high priority stages are taken into account. For example, consider a scenario tree similar to scenario tree (b) in figure 2, where the current tree includes only scenarios S1 and S2. For the next iteration, we can select among scenario S3 and *z*. By adding scenario S3, a node corresponding to stage 2 will be considered while by adding scenario z a node corresponding to stage 3 with a lower priority will be added to the scenario tree. So, based on the *stage priority* rule, S3 should be selected as the candidate scenario.

In fact, as the decision variables corresponding to initial stages (high priority ones) are fixed for the following stages, if we consider the majority of nodes in those stages, as we proceed in the algorithm, the impact of adding new scenarios on the objective function value is decreased gradually. This can accelerate the convergence of the algorithm into a good solution. As in many industries (e.g., in sawmills) the production plan is usually updated based on a rolling horizon, the idea is to obtain a production plan for the majority of scenarios (nodes) in the initial stages of the scenario tree (the initial periods). Moreover, mixing the *stage priority* rule with the *highest probability* rule properly, a more balanced approximate scenario tree is resulted, comparing to the rules proposed in Lulli and Sen (2006). A more balanced approximate scenario tree can be helpful to avoid too optimistic or pessimistic approximate solutions.

Regarding the large dimensionality of our problem, adding only one scenario increases considerably the complexity of the resulting MS-MIP. Thus, in the updating phase, we add one scenario with the largest change in the objective function value per iteration of the algorithm. Note that the current solution of the problem should be used as a warm start for the next iteration of the algorithm.

6- Computational results

In this section, we provide the implementation results of the two solution strategies for the sawmill production planning example. CPLEX 11 and OPL 6.1 are used to implement the PHA and the RSU method. All numerical experiments are conducted on an AMD Athlon(tm) 64×2 dual core processor 3800+, 2.01 GHz, 1.00 GB of RAM, running Microsoft Windows Server 2003, standard edition.

Solution strategy 1 (the progressive hedging algorithm)

As it was mentioned in section 4, at iteration 0 of the PHA, the scenario MIP sub-problems are solved to find good integer solutions. Each scenario sub-problem at iteration 0 includes 22,500 constraints and 44,341 decision variables. Good integer solutions which are very close to the optimal one are found by considering a 3000 seconds global time limit for the MIP solver in CPLEX 11. It should be noted that the MIP solver of CPLEX 11 stops in about 17000 seconds to find an optimal integer solution with a very small optimality gap. However, this solution is not significantly different with the best integer solution found after 3000 seconds. The implementable solutions are computed for the nodes of the demand scenario tree at stages 2 and 3 by averaging the solutions for corresponding scenarios. In the following iterations, the penalty terms are added into the objective function of each scenario sub-problem to hedge against the non-anticipatipativity. Each scenario MIQP sub-problem includes 23,180 constraints and 45,021 decision variables. The MIQP solver of CPLEX 11 is used to find the optimal solution, which results the optimum in about 2000 seconds. It was shown in Løkketangen and Woodruff (1996) that the PHA is not sensitive to parameter r values. By considering $r=0.6$, the PHA converges for all integer variables after 3 iterations, which is quite fast. Thus, we did not consider larger values of r. After fixing the binary variables in the deterministic equivalent model at their converged values, this model is solved for continuous decision variables. The optimal solution has an objective function value (cost) equal to 2,565,031. As it was mentioned in section 4, the converged solution of the PHA in a non-convex case is a local optimal one for the MS-MIP model, thus its objective value can be considered as an upper bound for the optimal solution.

It would be worth mentioning that we also examined a tabu search (TS) algorithm for general MIP models with the search on integer variables (see for example, Pedroso, 2005; Huang et al., 2003) to solve the MIP and MIQP scenario sub-problems in the PHA, instead of solving them by

CPLEX. However, the TS did not succeed to converge into a good solution faster than CPLEX11, since at each iteration a number of relaxed large-scale LP and QP models should be solved to optimality.

The solution strategy 2 (the revised scenario updating method)

The algorithm is initiated by considering 3 scenarios (among 27) based on the *dissimilarity* rule (i.e., three scenarios with the highest, average and the lowest objective function values are selected). At the following iterations, the candidate scenario is selected based on the *stage priority* and the *highest probability* rules by putting more emphasis on the *stage priority* rule. It should be mentioned that for each scenario sub-tree in each iteration of the RSU algorithm the deterministic equivalent model represented by compact formulation is solved by the MIP solver of CPLEX 11. Table 1 summarizes the results of scenario updating method for the sawmill example. The third column in table 1 indicates the size of model (8)-(14) for each scenario sub-tree and column 4 corresponds to its objective function value. Column 5 indicates the estimated upper bound on the objective function value by adding the candidate scenario selected based on the mentioned rules. The sixth column corresponds to the relative gap between the estimated upper bound and the current objective function value. Positive values in this column indicate a decline in the objective function value while the negative gaps indicate an improvement, after adding the candidate scenario. As it can be observed from table 1, as the algorithm proceeds, the gap between the expected cost of the new sub-tree by the current one decreases. Finally, at iteration 5, the values of this gap computed for 4 candidate scenarios (with the highest stage priority and probability) are negligible. Thus, it can be concluded that adding more scenarios to the current scenario tree would not affect the objective function value significantly and the current solution can be considered as a good approximate one. It would be worth mentioning that at iteration 5, 100%, 78% and 27% of nodes corresponding to stages 2, 3 and 4 in the scenario tree described in section 3 are taken into account. Moreover, at initial iterations, we found an approximate solution for the reduced model in step 1 of the algorithm by limiting the global time of the MIP solver in CPLEX. Only at the last iteration, the problem is solved for the optimal solution.

It should be mentioned that by using the scenario selection rules in Lulli and Sen (2006) the algorithm did not convergence until iteration 6.

Insert table 1 here.

The obtained results demonstrate the efficiency of the applied scenario selection rules (stage priority and the highest probability) in the RSU method in convergence after a quite small number of iterations. The comparison between the approximate solution of the RSU method with the local optimal solution of the solution strategy 1 (PHA), indicates a 14% gap between this approximate solution and the upper bound of the optimal objective value. Regarding that the approximate solution is obtained by considering less than 30% of scenarios and 50% on nodes in the original scenario tree, it can be considered as a good solution which can be obtained considerably faster than a local optimal solution (by PHA).

Finally, we believe that the approximate solution of the RSU method in this example can be considered as a lower bound for the optimal solution. Using the stage priority and the highest probability rules for scenario selection in the RSU method, only 2 nodes are not considered at stage 3 (a low demand after an average demand at stage 2 and a high demand after a low demand at stage 2). Thus, we can conclude that the obtained solution is quite realistic till stage 3. However, since at stage 4 only 25% of scenarios (nodes) are taken into account we can expect that the real expected cost (the optimal objective value) be higher than the cost obtained by RSU method.

7- Conclusion

In this paper, we used two solution strategies to find good solutions for a large-scale multi-stage stochastic mixed-integer production planning problem under uncertainty in products demands and processes yields. The deterministic equivalent model by considering all scenarios for random yield and demand could not be solved by CPLEX, regarding its large dimension. The first algorithm is based on the progressive hedging algorithm (PHA) which is used for the integer variable convergence. The scenario sub-problems are solved for optimality by CPLEX 11. The obtained solution is an upper bound for the optimal objective value. The second algorithm is a scenario updating method (Lulli and Sen, 2006) revised in two directions. First, we verified the efficiency of bounds proposed in the algorithm for MS-MIP models with full recourse. We used only the proposed upper bound in the algorithm. Moreover, we modified the scenario selection rules and we proposed a new rule to improve the rate of convergence and the quality of the solution. As the case study, production planning under the uncertainty in raw material (log) quality and product (lumber) demand in a realistic scale prototype sawmill was considered. The computational results indicated that the revised scenario updating algorithm converges in quite small number of iterations. Moreover, the approximate solution which is obtained by considering a small portion of scenarios in the original scenario tree has an acceptable gap with the upper bound found by the PHA.

The revised scenario updating method should be applied for more examples of multi-stage stochastic MIP models with different scenario trees for a complete validation of the proposed modifications. Moreover, heuristic algorithms which are able to find good integer solutions for general (without a special structure) MIP and MIQP scenario sub-problems should be developed and applied in the PHA to reduce the execution time of this algorithm. Implementing the PHA on parallel machines can also reduce significantly the running time.

Acknowledgment

This work was supported by FOR@C research consortium of Université Laval.

References

Bakir, M.A., and Byrne, M.D., 1998. Stochastic linear optimization of an MPMP production planning model. International Journal of Production Economics 55, 87-96.

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

Brandimarte, P., 2006. Multi-item capacitated lot-sizing with demand uncertainty. International Journal of Production Research 44(15), 2997-3022.

Cranic, T.G., Fu, X., Gendreau, M., Rei, W., and Wallace, S.W., 2009. Progressive hedging – based meta-heuristics for stochastic network design. CIRRELT-2009-03 working document.

Dupačová, J., 1995. Postoptimality for multi-stage stochastic linear programs. Annals of Operations Research 56, 65-78.

Escudero, L.F., Kamesam, P.V., King, A.J., and Wets, R.J-B., 1993. Production planning via scenarios. Annals of Operations Research 34, 311- 335.

Haugen, K.K., Løkketangen, A., and Woodruff, D.L., 2001. Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. European Journal of Operational Research, 132, 116-122.

Huang, K., 2005. Multi-stage stochastic programming models for production planning. Dissertation, School of Industrial and Systems Engineering, Georgia Institute of Technology.

Hung, Y-F., Chen, CH-P., Shih, CH-CH., and Hung, M-H., 2003. Using tabu search with candidate list to solve production planning problems with setups. Computers & Industrial Engineering, 42, 615-634.

Kazemi Zanjani, M., Ait-Kadi, D., and Nourelfath, M., 2010a. Robust production planning in a manufacturing environment with random yield: A case in sawmill production planning. European Journal of Operational Research 201, 882-891.

Kazemi Zanjani. M., Nourelfath, M., and Ait-Kadi, D., 2010b. A multi-stage stochastic programming approach for production planning with uncertainty in the quality of raw materials and demand. International Journal of Production Research, doi: 10.1080/00207540903055727.

Kazemi Zanjani, M., Nourelfath, M. and Ait-Kadi, D., 2010c. Production planning with uncertainty in the quality of raw materials: A case in sawmills. Journal of Operations research Society, doi: 10.1057/jors.2010.30.

Klein Haneveld, W.K., and van der Vlerk, M.H., 1999. Stochastic integer programming: General models and algorithms. Annals of Operations Research 85, 39-57.

Leung, S.C.H., Tsang, S.O.S., Ng, W.L., and Wu, Y., 2007. A robust optimization model for multi-site production planning problem in an uncertain environment. European Journal of Operational Research 181(1), 224-238.

Løkketangen, A. and Woodruff, D.L., 1996. Progressive hedging and tabu search applied to mixed-integer (0,1) multi-stage stochastic programming. Journal of heuristics 2, 111-128.

Lulli, G. and Sen, S., 2004. A branch-and-price algorithm for multi-stage stochastic integer programming with application to stochastic batch-sizing problems. Management Science 50, 786-796.

Lulli, G. and Sen, S., 2006. A heuristic procedure for stochastic integer programs with complete recourse. European Journal of Operational Research 171, 879-890.

Miller, A.C. and Rice, T.R., 1983. Discrete approximations of probability distributions. Management Science 29(3), 352-362.

Mula, J., Poler, R., Garcia-Sabater, J.P. and Lario, F.C., 2006. Models for production planning under uncertainty: A review. International Journal of Production Economics 103, 271-285.

Mulvey, J.M., Vanderbei, R.J. and Zenios, S.A., 1995. Robust optimization of Large-scale systems. Operations Research 43(2), 264-281.

Pedroso, J.P., 2005. Tabu search for mixed-integer programming. In: Sharda et al., Metaheuristic Optimization via Memory and Evolution, chapter 11, 247-261.

Rockafellar, R.T. and Wets, R.J-B., 1991. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research 16(1), 119-147.

Schultz, R., 2003. Stochastic programming with mixed-integer variables. Mathematical programming 97, 285-309.

Sen, S., 2003. Algorithms for stochastic mixed-integer programming models. Stochastic Integer Programming Handbook, Dordrecht.

Figures

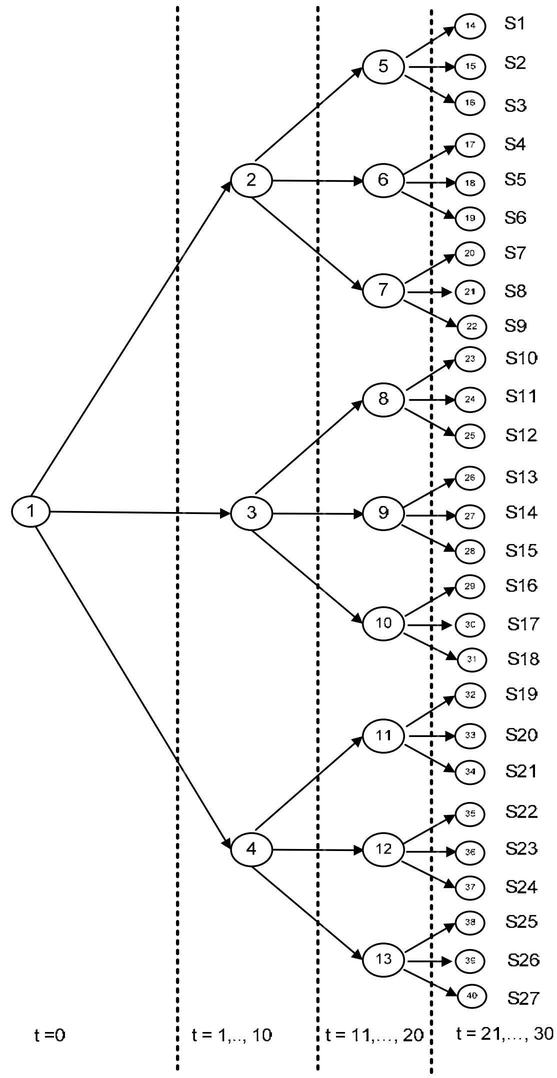

Figure 1. Scenario tree for the random demand

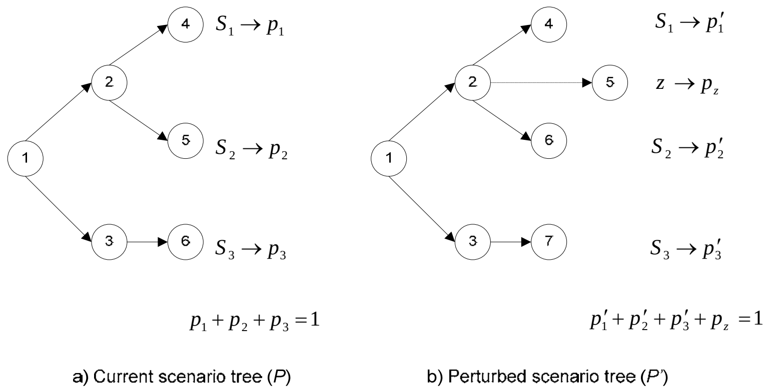

Figure 2- Perturbation of a scenario tree by a single scenario *z*

Tables

Table 1: Results of the revised scenario updating method

Iteration	#scenarios	#constraints/#decision variables	Objective function value	The upper bound	The relative gap (%)	Scenario selection rule
1	3	60,000/118,240	2,383,544	2,269,416	-5	*dissimilarity*
2	4	82,500/162,580	2,097,401	2,204,807	4.8	*stage priority + highest probability*
3	5	97,500/192,140	2,154,496	2,252,988	4.3	*stage priority + highest probability*
4	6	112,500/221,700	2,241,741	2,204,618	-1.7	*stage priority + highest probability*
5	7	127,500/251,260	2,186,905	2,186,087 ($\pm$ 0.14%)*	-0.04	*stage priority + highest probability*

*The average and standard deviation for 4 candidate scenarios