Available online at www.sciencedirect.com
SCIENCE DIRECT®
Int. J. Production Economics 101 (2006) 140–149
international journal of production economics
www.elsevier.com/locate/ijpe

Scenario-based planning for lot-sizing and scheduling with uncertain processing times

Patrizia Beraldi[a,*], Gianpaolo Ghiani[b], Emanuela Guerriero[c], Antonio Grieco[b]

[a]Dipartimento di Elettronica, Informatica e Sistemistica, Università degli Studi della Calabria, 87036 Rende (CS)–Italy
[b]Dipartimento di Ingegneria dell'Innovazione, Università degli Studi di Lecce, 73100 Lecce, Italy
[c]Dipartimento di Economia e Statistica, Università degli Studi della Calabria, 87036 Rende (CS)–Italy

Available online 11 October 2005

Abstract

This paper addresses the identical parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs and uncertain processing times. The evolution of the uncertain parameters is modelled by means of a scenario tree, giving rise to a multistage stochastic mixed-integer program. Fix-and-relax procedures, exploiting the specific structure of the problem, are developed and compared. Computational results on a large set of randomly generated instances show that the gap between the best heuristic solutions and the lower bounds provided by a truncated branch-and-bound never exceeds 3%.

Keywords: Stochastic programming; Lot-sizing and scheduling problems; Heuristics; Textile industry; Fiberglass industry

1. Introduction

Lot-sizing and scheduling problems have been an area of intensive research activity starting from the seminal paper of Wagner and Whitin (1958). Since then, there have been several contributions aimed at incorporating real-world features (such as backlogging, production capacities, multiple items, multiple machines and multiple stages) into the basic model. We refer the reader to the surveys of Wolsey (1997), Drexel and Kimms (1997), Staggemeier and Clark (2001) and to the recent up-to-date tutorial of Pochet (2001).

Lot-sizing and scheduling problems involving set-ups play a key role in many industries. Whenever a set-up only affects a machine idle time, set-up costs are directly proportional to set-up times in which case schedules optimal with respect to set-up times are also optimal with respect to set-up costs. On the other hand, when

*Corresponding author. Tel.: +39 0984 494826; fax: +39 0984 494713.
E-mail addresses: beraldi@unical.it (P. Beraldi), gianpaolo.ghiani@unile.it (G. Ghiani), emanuela.guerriero@unile.it (E. Guerriero), antonio.grieco@unile.it (A. Grieco).

0925-5273/$ - see front matter © 2005 Published by Elsevier B.V.
doi:10.1016/j.ijpe.2005.05.018

set-ups require skilled workforce, set-up costs are typically high while set-up times may be low. See (Allahverdi et al., 1999; Pinedo, 1995) for in-depth reviews on this topic.

While deterministic lot-sizing and scheduling problems have received significant coverage in the literature, their stochastic counterparts have been addressed only recently. See (Sox et al., 1999) for a survey of the current research literature on stochastic lot scheduling problems.

This paper deals with a stochastic version of the lot-sizing and scheduling problem with identical parallel machines and sequence-dependent set-up costs (LSIPMSC) in which processing times are uncertain. In the LSIPMSC problem, a number of identical parallel machines have to manufacture a set of products over a discrete planning horizon. The problem consists of determining the schedule that minimizes the total set-up cost while satisfying the demand of each product over the time horizon. Applications of the LSIPMSC arise in several industrial settings, e.g. in the textile and fiberglass industry (Dearing and Henderson, 1984).

The deterministic LSIPMSC problem, which is NP-hard, is addressed in a number of papers: Clark and Clark (2000), Meyer (2002), Staggemeier et al. (2002) examine the case of heterogeneous machines whereas Beraldi et al. (submitted) studies the case of homogeneous machines. However, to our knowledge, all the efforts have been concentrated on models where input data are assumed deterministically known. This assumption is rather restrictive since in real-world different sources of uncertainty (e.g. processing times, product demands, etc.) may affect lot-sizing and scheduling decisions. As a consequence, the solution provided by a deterministic model where the uncertain parameters are replaced by either mean or worst case estimations may be of little value.

In this paper we use a multistage stochastic programming framework to explicitly deal with uncertain processing times. In the last few decades, stochastic integer programming has been the subject of intense research activity. However, most papers deal with two-stage programs while very few papers address the multi-stage case. See Sen (2003) and the references listed at http://mally.eco.rug.nl/spbib.html, for an updated survey on stochastic programming. Also see (Lulli and Sen, to appear) for a list of references on economic lot-sizing models under uncertainty.

In this paper we present novel optimization-based heuristics for the stochastic LSIPMSC problem, based on the fix and relax framework introduced by Dillenberger et al. (1994) in a deterministic setting. The main idea of this approach is to decompose the original problem into a sequence of subproblems where the requirement of integrality is imposed on a limited number of variables. The remainder of the paper is organized as follows. Section 2 presents a multi-stage stochastic integer programming formulation of the LSIPMSC problem with uncertain processing times. Section 3 is devoted to the presentation of some heuristic strategies. The performance of the fix-and-relax procedures has been numerically tested on a variety of medium and large size problems. The results are reported in Section 4. Finally, conclusions and future research directions follow in Section 5.

2. Problem formulation

In this section a multi-stage stochastic programming formulation of the LSIPMSC problem is presented. We first outline a formulation for the deterministic LSIPMSC (Beraldi et al., submitted), then we illustrate a formulation of the LSIPMSC with uncertain processing times.

2.1. Deterministic LSIPMSC

In the LSIPMSC problem, M identical parallel machines have to manufacture N products over a discrete planning horizon of T periods. Each product requires a single machining operation and no pre-emption is allowed. A changeover is associated to a (sequence dependent) set-up cost but causes no relevant machine idle time. Moreover, at each period $t = 1, \ldots, T$, product demands must be satisfied (i.e., no backlog is allowed). The aim is to allocate products to machines over time in such a way that product demands are satisfied at any time and the total set-up cost is minimized.

Let a_{it} and d_{it} represent the production rate and the demand of product i at time period t, respectively. Furthermore, let c_{ijt} be the cost paid to switch from product i to j at time period t. The problem can be formulated as follows:

$$\text{Min} \quad \sum_{t=1}^{T}\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} c_{ijt}y_{ijt}, \tag{1}$$

$$\text{s.t.} \quad x_{it-1} + \sum_{j=1, j\neq i}^{N} y_{jit} - \sum_{j=1, j\neq i}^{N} y_{ijt} = x_{it},$$

$$i = 1, \dots, N; \;\; t = 1, \dots, T, \tag{2}$$

$$\sum_{l=1}^{t} a_{it}x_{it} \geqslant \sum_{l=1}^{t} d_{it},$$

$$i = 1, \dots, N; \;\; t = 1, \dots, T, \tag{3}$$

$$x_{it} \geqslant 0 \quad \text{and integer},$$

$$i = 1, \dots, N; \;\; t = 1, \dots, T, \tag{4}$$

$$y_{ijt} \geqslant 0, \quad i,j = 1, \dots, N; \;\; t = 1, \dots, T. \tag{5}$$

Variable x_{it} denotes the number of machines assigned to product i at time period t, whereas y_{ijt} represents the number of machines switched from product i to product j at time period t. Variables x_{i0} are equal to M_i ($\sum_{i=1}^{N} M_i = M$). Constraints (2) set the number of changeovers while constraints (3) state that backlogging is forbidden. It can be shown that, provided that parameters a_{it} and d_{it} are integer, the integrality restrictions on y_{ijt} variables can be eliminated. See (Beraldi et al., submitted) for more details.

2.2. LSIPMSC with uncertain processing times

In what follows, processing times are assumed to be uncertain because of machines' production rates variability. On the other hand, orders are supposed to be negotiated at the beginning of the planning horizon so that product demands are deterministic. We assume that each uncertain parameter evolves as a discrete time stochastic process with a finite probability space. We may represent the uncertain information as a multi-layered tree, where nodes n at level t ($t = 1, \dots, T$) represent the state of the production system up to stage t (Ahmed et al., 2003). Each node n of the scenario tree, except the root ($n = 0$), has a unique parent $b(n)$, and each non-terminal node n is the root of a sub-tree $\Gamma(n)$. S_t denotes the set of nodes corresponding to time stage t, as well as t_n and p_n are the time stage and the probability level corresponding to node n, respectively ($\sum_{n\in S_t} p_n = 1$ for each $t \in \{1, \dots, T\}$). The path from the root node to a node n is denoted by $Q(n)$. If n is a terminal node (leaf), then $Q(n)$ corresponds to a scenario and represents a joint realization of uncertain parameters over all periods $1, \dots, T$. Assuming that there are S leaf nodes corresponding to S scenarios, Q_s will denote the path corresponding to scenario s, with $s = 1, \dots, S$. Then, the stochastic LSIPMSC problem (in the following referred to as P for short) can be formulated as follows:

$$\text{Min} \sum_{n\in\Gamma(0)}\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N} p_n c_{ijn} y_{ijn}, \tag{6}$$

$$x_{ib(n)} + \sum_{j=1, j\neq i}^{N} y_{jin} - \sum_{j=1, j\neq i}^{N} y_{ijn} = x_{in},$$

$$i = 1, \dots, N; \;\; n \in \Gamma(0), \tag{7}$$

$$\sum_{m\in Q(n)} a_{im}x_{im} \geqslant \sum_{m\in Q(n)} d_{it_m}, \quad i = 1, \dots, N;$$

$$n \in \Gamma(0), \tag{8}$$

$$x_{in} \geqslant 0 \text{ integer}, \quad i = 1, \dots, N; \;\; n \in \Gamma(0), \tag{9}$$

$$y_{ijn} \geqslant 0, \quad i,j = 1, \dots, N; \;\; n \in \Gamma(0), \tag{10}$$

where $x_{ib(0)} = M_i$. The model represented by Eqs. (6)–(10) is a multistage stochastic mixed-integer programming model where the non-anticipativity constraints are implicitly included in the formulation. They state that all the scenarios with the same history until a given stage should result in the same decisions until that stage. Formulation (6)–(10) is a deterministic mixed integer program, which, in principle, can be solved to optimality by using general integer programming techniques such as dynamic programming, branch-and-bound, branch-and-cut or branch-and-price. However, for practical sized instances, such approaches are often unable to converge or even fail to provide a feasible solution in a reasonable amount of time. This motivates the need of efficient heuristic methods. In the following section, we describe some optimization-based heuristic strategies, tailored to the stochastic LSIPMSC problem.

3. Fix-and-relax procedures

Heuristic procedures for stochastic programming problems have been proposed only recently. A first contribution is due to Løkketangen and Woodruff (1996) who have proposed a combination of the progressive hedging algorithm with a tabu search approach. Also see (Haen et al., 2001). More recently, Ahmed and Sahinidis (2003) have described a heuristic strategy based on the idea of decomposing the original problem into a sequence of deterministic scenario subproblems, each approached heuristically. The non-anticipativity condition is recovered by a capacity bundling procedure. This method was successfully applied to a capacity expansion problem.

The main idea behind our heuristic approach is to enforce the non-anticipativity condition and solve a sequence of subproblems where only a limited number of variables is required to be integer. The scheme is based on the fix and relax (FR) approach proposed in Dillenberger et al. (1994) for the solution of deterministic production scheduling problems (also see (Escudero and Salmeron, to appear)). Such an approach has been applied, in its basic version, in Alonso et al. (2000) for the solution of a stochastic 0–1 air traffic flow management problem. In what follows, we present an enhancement of the basic FR version (MSFR) which overcomes the main drawbacks of the original scheme. The MSFR approach solves a sequence of k mixed-integer subproblems P^r $(r = 1, \dots, k)$, associated to the nodes of the scenario tree. Let V be the set of node indices of the scenario tree $\Gamma(0)$ and let V_p be a partition of V into k subsets. Moreover, q_p denotes the cardinality of V_p $(|\Gamma(0)| = \sum_{p=1}^{k} q_p)$. Subproblem P^r $(r = 1, \dots, k)$ is obtained by fixing the values of some variables and relaxing the integrality condition on the other variables. In particular, the integrality constraints (10) is replaced with

$$x_{in} = \hat{x}_{in} \quad \forall n \in V_p,\ i = 1, \dots, N,\ p = 1, \dots, r-1 \text{ (if } r > 1), \tag{11}$$

$$x_{in} \in Z^+ \quad \forall n \in V_r,\ i = 1, \dots, N, \tag{12}$$

$$x_{in} \geqslant 0 \quad \forall n \in V_p,\ i = 1, \dots, N,\ p = r+1, \dots, k \text{ (if } r < k). \tag{13}$$

Hence, at each iteration $r > 1$, the set of variables is partitioned in three subsets: (1) the set of "fixed" variables (those variables whose values are given by the solution $\hat{x}_{in}$ for $n \in V_p$, $i = 1, \dots, N$, $p = 1, \dots, r-1$ of subproblems $P^1, \dots, P^{r-1}$); (2) the set V_r of variables required to be integer; (3) the set of "relaxed" variables (those variables associated to nodes belonging to subsets V_p not yet processed). Solving subproblems P^r is relatively easy since at each iteration only a limited number of variables is required to be integer.

Let z_r^* and $\bar{z}$ be the optimal objective function value of P^r and an upper bound on the optimal solution of the original problem, respectively. More formally, the MSFR strategy is as follows:

Step 0. *Initialization*: Set $r = 1$ and solve P^1. If P^1 is infeasible, **STOP**: the original problem is infeasible.

Step 1. *Termination*: If $r = k$, **STOP**: the problem is feasible. Set $\bar{z} = z_r^*$. Otherwise, increase r by 1.

Step 2. *Solution*: Solve P^r. If a feasible solution is found, go to Step 1.

Step 3. *Backwards grouping step*: Redefine the partition structure as following:

$$\begin{aligned} &V_{r-1} \longleftarrow V_{r-1} \cup V_r, \\ &V_i \longleftarrow V_{i+1}, \quad \forall i = r, \dots, k-1, \\ &k \longleftarrow k-1. \end{aligned} \tag{14}$$

Decrease r by 1. If $r = 1$ go back to Step 0. Otherwise, go back to Step 1.

It is worthwhile noting that, if the original problem is feasible, the MSFR algorithm terminates in a finite number of iterations with a feasible solution. Indeed, infeasibility is prevented by Step 3, which merges two or more subsets of the original partition in case of infeasibility at the previous stage. Of course, this can result in solving the original problem. Under this respect, this feature might be inefficient (Alonso et al., 2000). In order to overcome this drawback, we have developed two variable partitioning policies tailored to the stochastic LSIPMSC problem.

3.1. Time partitioning policy

A fundamental issue of the MSFR approach is the definition of the partitioning policy of the set V. Because of the structure of the scenario tree, the most natural choice is a time-stage partition. We have set k equal to the number T of stages of the scenario tree and we have assigned to each node n a value w_n equal to t_n. Each subset V_r is defined as

$$V_r = \{n \in \Gamma(0) | w_n = r\} \quad \forall r = 1, \ldots, T, \tag{15}$$

where

$$\forall r, r' \in [1, T] \quad r \leqslant r' \;\Leftrightarrow\; w_r \leqslant w_{r'}. \tag{16}$$

The algorithm first determines the values of the variables associated with early time-stages, assuring the satisfaction of the non-anticipativity condition. At iteration r ($r = 1, \ldots, T$), the number of integer variables depends on the number of nodes in the rth level of the scenario tree. Problem P^r can be decomposed into $|V_r|$ independent subproblems. Thus, for each level of the scenario tree we solve a limited number of independent subproblems where the integrality condition is imposed only on a subset of variables. However, the proposed scheme does not prevent the infeasibility of the subproblems, hence requiring the execution of the backtrack step from time to time. In the following subsection we introduce an enhanced approach which prevents this undesired algorithmic behavior.

3.2. Enhanced subproblem formulation

Let $\Psi = \{\xi_q\}_{q=1}^{S}$, where $\xi_q = \{\gamma_n\}_{n \in P_q}$ and let $\gamma_n = \{a_{in}\}_{i=1}^{N}$ be the set of scenarios. In some cases it is possible to define a partial order relation $\preceq$ on γ_n, in case scenario q is "not harder" than scenario q'. We have

$$q \preceq q' \;\Leftrightarrow\; \xi_q \geqslant \xi_{q'}. \tag{17}$$

By using this partial order relation, it is possible to classify the set of scenarios identifying the most representative ones, which can be used in the definition of the solution approach. A *guiding scenario* ξ_i is such that $\xi_i \leqslant \xi_q$ for each $q = 1, \ldots, S, q \neq i$. As a rule, we may identify a guiding scenario not for the entire tree, but for a sub-tree $\Gamma(n)$ rooted at a particular node n. We shall refer to such scenarios as *partial guiding scenarios*. In the following, we use the attribute *consistent* to identify a scenario tree for which guiding scenarios, either defined on the entire tree or on sub-tree, may be identified.

For each product i, we denote by g_{it}^n the minimum production rate at time stage t:

$$g_{it}^n = \min(a_{im} | m \in S_t(n)), \tag{18}$$

where $S_t(n)$ indicates the set of nodes in $\Gamma(n)$ corresponding to time stage t, with $t = t_n, \ldots, T$. These values may be used to define a far-sighted policy: the decision at a given node of a level should be taken in such a way to hedge all the situations that can happen in the future. From a mathematical point of view, this can be accomplished by adding new constraints and variables. Let us consider the nodes $n \in V_r$. The enhanced formulation is defined by adding the following constraints to each subproblems P_n^r:

$$\forall i \in [1, N], \quad \forall t \in [t_n + 1, T]: \tag{19}$$

$$a_{in} x_{in} + \sum_{t=t_n+1}^{T} g_{it}^n u_{it}^n \geqslant \sum_{m \in Q(n)} d_{im} - \sum_{m \in P(b(n))} a_{im} \widehat{x}_{im}, \tag{20}$$

$$\sum_{i=1}^{N} u_{it}^n = M \quad \forall t \in [t_n + 1, T], \tag{21}$$

$$u_{it}^n \in Z^+ \quad \forall i \in [1, N], \quad \forall t \in [t_n + 1, T]. \tag{22}$$

Constraints (20)–(22) refer to a dummy scenario ζ_n corresponding to the worst machine production rate values (i.e. $\zeta_n = (\zeta_t^n)_{t \in [t_n, T]}$, where $\zeta_t^n = (g_{it}^n)_{i=1}^{N}$). Therefore, in the enhanced formulation each subproblem P_n^r is defined over a consistent scenario sub-tree obtained by extending Ψ_n with a dummy guiding scenario ζ_n. We account for this scenario by means of variables u which are not included in the objective function. The overall effect is to preserve some production in order to hedge future bad situations that can occur. We observe that if Ψ_n is consistent, then constraints

(20)–(22) are equivalent to impose the integrality restriction on the decision variables corresponding to the partial guiding scenario, hence avoiding the introduction of additional variables.

The following theorem (whose proof is omitted for the sake of brevity Guerriero) states that the enhanced formulation prevents infeasibility, at least if the problem itself is feasible.

Theorem 3.1. *If problem* (6)–(10) *is feasible, partition* $V_1, \ldots, V_k$ *is obtained according to the TP policy and subproblems* P^r *are defined according to enhanced formulation, then all subproblems* P^r *are feasible.*

3.3. Further issues

The approach illustrated in the previous section prevents infeasibility at the expense of an increase in the number of variables and constraints. Let $|V_r|$ be the number of nodes of the rth partition. It is evident that for each problem P_n^r the only variables to be considered are those related to nodes contained in the sub-tree rooted at n. This observation allows the solution phase to be fully split by considering as many independent subproblems as the cardinality of V_r. The enhanced formulation guarantees feasibility of the problem by taking future into account by means of pessimistic estimates. In particular, for each product and each future stage an integer variable is included. Thus, the total number of integer variables added is proportional to $N \times (T - r)$, where $(T - r)$ denotes the difference between the last and the current level of the scenario tree. Such variables appear in $(T - r) + N \times (T - r)$ constraints. Because of the proportionality to r, the corresponding problems contains a lower number of additional variables and constraints, as we proceed in the scenario tree. Nevertheless, even for $r = 1$ such number is rather limited. As a matter of fact, this corresponds imposing the integrality condition on the variables associated with a single scenario.

The effectiveness of the proposed approach may be improved by trying to reduce the number of extra integer variables and constraints included. This can be accomplished by defining clusters of levels, i.e grouping nodes belonging to different levels and imposing the integrality restriction on the decision variables related to the selected nodes. The effect produced by this policy is an improvement of the lower bound value provided by the solution of the subproblems associated to the nodes of the partition.

On the basis of the information structure of the scenario tree, we have defined two different selection policies. The first one, referred to as TPH, consists of defining a H-width cluster including nodes related to time stages in $[r + 1, r + H]$. The second one, referred to as MPS, consists of selecting nodes belonging to the most probable scenario in the current sub-tree. In order to preserve the computational tractability of each subproblem, the selected nodes are sorted on the basis of time stages and probability level and only the first Q nodes are, finally, included into the cluster. Parameter Q has been chosen so that the corresponding total number of integer variables does not exceed a given threshold F (i.e. $Q * N \leqslant F$).

4. Computational results

The solution strategies described in Section 3 have been coded in $C++$ language and linked to the CPLEX 8.1 library. Computational tests have been performed on a set of randomly generated instances whose characteristics are similar to the instances encountered in the textile and fibreglass industries. Indeed, the number of machines M has been set equal to 30, while the number of products N has been generated between 6 and 10. The length of the planning horizon has been varied between 4 and 7, and the set-up costs have been randomly generated in the range [1,10]. For each product, production rates $\{a_{it}\}_{t=1}^{T}$ have been modelled as a sequence of i.i.d. random variables with bounded support $[U_i, B_i] \cap Z^+$. The lower and upper bounds of the support are reported in Table 1.

The production rate uncertainty is due to human worker requirement. Upper bounds are determined on the basis of technological aspects. Lower bounds are considered greater than zero.

Indeed, the event corresponding to zero machine production rate represents the complete failure of all machines. Since there are tens of machines in the production environment, this event has been considered with null probability level. Moreover, we refer to a production technology (i.e. textile industries), where repairing time are negligible respect to planning period and therefore the number of available machine does not change during the planning horizon.

Table 1
Production rate ranges

Product i	1	2	3	4	5	6	7	8	9	10
U_i	90	180	324	162	90	126	234	108	324	270
B_i	130	260	467	233	130	182	337	156	467	389

Table 2
Instance features

Instance types	T	N	$\|\Gamma(0)\|$	S	x_{in} variables	y_{ijn} variables
$P_{4,6}$	4	6	15	8	90	540
$P_{4,8}$	4	8	15	8	120	960
$P_{4,10}$	4	10	15	8	150	1500
$P_{5,6}$	5	6	31	16	186	1116
$P_{5,8}$	5	8	31	16	248	1984
$P_{5,10}$	5	10	31	16	310	3100
$P_{6,6}$	6	6	63	32	378	2268
$P_{6,8}$	6	8	63	32	504	4032
$P_{6,10}$	6	10	63	32	630	6300
$P_{7,6}$	7	6	127	64	762	4572
$P_{7,8}$	7	8	127	64	1016	8128
$P_{7,10}$	7	10	127	64	1270	12,700

The main features of the instances generated this way (tree size, number of scenarios, number of integer and continuous variables) are reported in Table 2. For each instance type, both consistent (C) and inconsistent (NC) scenario trees have been generated. Moreover, two cases have been considered: (i) due-dates are uniformly distributed over the planning horizon (NE); (ii) all due dates are equal to T(E). For each combination of these features, five instances were generated. All the computations have been carried on a personal computer clocked at 800 MHz and equipped with a 256 MB RAM. Four variants of the F&R approach have been tested. Three of them (HEU_1, HEU_2 and HEU_3) are based on the time partitioning scheme with a forecast window length equal to 0, 1 and 2, respectively. The fourth procedure (HEU_4) implements the most probable scenario selection policy. In both HEU_1, HEU_2, HEU_3 and HEU_4 procedures, an upper bound $Q = 30$ was imposed on the number of integer variables in each subproblem. For most instances, general purpose mixed integer solver Cplex 7.0 was unable to determine the optimal solution within a reasonable amount of time. Therefore, the perfor-

Table 3
Computational results (equal due dates—consistent scenario tree case)

Instance type	$\text{GAP}_{\text{HEU}_1}$		$\text{GAP}_{\text{HEU}_2}$		$\text{GAP}_{\text{HEU}_3}$		$\text{GAP}_{\text{HEU}_4}$	
	Average	Max	Average	Max	Average	Max	Average	Max
$P_{4,6}$	0.45	3.74	0.24	4.13	0.47	1.8	0.88	1.91
$P_{4,8}$	2.71	2.74	0.75	1.38	0.41	0.52	1.2	1.27
$P_{4,10}$	0	0.83	0	1.33	1.15	1.41	0	0.58
$P_{5,6}$	0.17	0.57	0.09	2.03	0.6	0.7	0.17	0.57
$P_{8,5}$	0.6	0.67	0	0	0.17	1.93	0.6	0.67
$P_{5,10}$	0	0.14	0.61	1.65	0	0.5	0	0
$P_{6,6}$	0	0.04	0.63	2.46	0.61	2.59	0.04	1.38
$P_{6,5}$	0.55	0.9	0.51	1.35	0	0.95	0.43	0.91
$P_{6,10}$	0.15	3.39	0	0.13	0.13	1.02	0.11	1.95
$P_{7,6}$	1.78	4.98	1.87	3.47	1.39	4.62	0	0
$P_{7,5}$	0.93	2	1.56	3.33	2.15	3.47	0	0
$P_{7,10}$	0.22	0.95	0	0	0.37	1.18	0.22	0.96

mances of F&R approaches have been compared with the best feasible solution provided by Cplex 7.0 within 3600 s. This truncated branch-and-bound algorithm provides a lower bound LB on z^*. Computational results are reported in Tables 3–6 and Figs. 1–2. In Tables 3–6 the effectiveness of F&R approach has been evaluated through the mean and the maximum value of the percentage error defined as follows:

$$\mathrm{GAP}_i = \frac{\mathrm{UB}_{\mathrm{HEU}_i} - \mathrm{LB}}{\mathrm{LB}}. \tag{23}$$

Computational results indicate that the heuristic approach always succeeds in the consistent scenario tree case, where the best performances have been obtained by the time selection policy either in the equal and unequal due date cases. On the other hand, in the inconsistent case, the most probable scenario policy always outperforms the other heuristics. The results also show that the F&R heuristics require less time than the truncated branch-and-bound. As a rule, the consistent case requires a larger computational effort than the inconsistent one. Moreover, in the case of incon-

Table 4
Computational results (equal due dates—inconsistent scenario tree)

Instance type	$\mathrm{GAP}_{\mathrm{HEU}_1}$		$\mathrm{GAP}_{\mathrm{HEU}_2}$		$\mathrm{GAP}_{\mathrm{HEU}_3}$		$\mathrm{GAP}_{\mathrm{HEU}_4}$	
	Average	Max	Average	Max	Average	Max	Average	Max
$P_{4,6}$	2.59	5.48	1.15	4.43	0	2.19	1.83	2.41
$P_{4,8}$	5.04	7.73	2.54	3.82	0.27	1.53	0.3	5.53
$P_{4,10}$	6.43	15.83	1.18	3.42	1.18	3.66	1.18	2.16
$P_{5,6}$	0.09	1.4	0.47	1.63	0.85	1.63	0.13	1.46
$P_{8,5}$	5.53	8.18	2.2	2.28	1.05	1.66	0	0
$P_{5,10}$	7.13	9.36	1.21	11.23	0	7.77	1.35	2.02
$P_{6,6}$	0.31	4.26	1.53	2.75	0.47	0.78	0	0
$P_{6,5}$	1.21	5.52	1.29	3.34	1.8	2.92	0	1.22
$P_{6,10}$	0.06	5.55	2.49	4.76	0.09	1.33	0	0
$P_{7,6}$	3.9	5.32	0.63	4.13	1.27	4.81	0	0
$P_{7,5}$	0.33	3.72	1.04	2.29	1.49	3.61	0	0
$P_{7,10}$	2.2	5.67	0	4.57	0.38	0.44	0	0.91

Table 5
Computational results (unequal due dates—consistent scenario tree case)

Instance type	$\mathrm{GAP}_{\mathrm{HEU}_1}$		$\mathrm{GAP}_{\mathrm{HEU}_2}$		$\mathrm{GAP}_{\mathrm{HEU}_3}$		$\mathrm{GAP}_{\mathrm{HEU}_4}$	
	Average	Max	Average	Max	Average	Max	Average	Max
$P_{4,6}$	0	0.05	0	0.99	0	1.27	0	0.24
$P_{4,8}$	0.01	0.67	0	0.58	0	0.85	0	0.67
$P_{4,10}$	0.27	2.86	0.06	0.49	0.09	0.5	0.29	0.36
$P_{5,6}$	0.24	1.27	0	1.07	0.75	0.9	0.07	1.27
$P_{8,5}$	0.15	1.08	0.55	0.6	0	0.83	0.09	1.38
$P_{5,10}$	0.1	2.16	0.02	0.03	0.02	0.33	0	0
$P_{6,6}$	0.22	1.35	0.25	1.98	0	0.37	0	0.34
$P_{6,5}$	0.09	0.58	0.04	0.75	0.09	0.41	0.09	0.47
$P_{6,10}$	0	0.1	0.99	1.29	0.08	1.52	0	0.22
$P_{7,6}$	0.09	2.86	0	0.7	0	0.49	0.12	0.2
$P_{7,5}$	0	0.46	0	2.74	0.46	0.54	0.11	8.3
$P_{7,10}$	0	0.05	0	0.4	0.01	1.41	0	0.05

Table 6
Computational results (unequal due dates—inconsistent scenario tree case)

Instance type	GAP_{HEU_1}		GAP_{HEU_2}		GAP_{HEU_3}		GAP_{HEU_4}	
	Average	Max	Average	Max	Average	Max	Average	Max
$P_{4,6}$	2.46	3.58	0.98	1.43	0.56	1.11	0	0.64
$P_{4,8}$	0.17	2.42	1.11	1.43	0.11	1.22	0.79	1.66
$P_{4,10}$	2.35	12	0.81	2.4	0.99	4.17	1.03	1.81
$P_{5,6}$	1.66	1.86	1.24	2.16	0.99	1.12	0	0
$P_{8,5}$	2.77	3.62	2.29	2.92	0.45	2.05	0	3.62
$P_{5,10}$	1.77	9.54	2.84	3.26	1.21	1.68	0	0
$P_{6,6}$	0	3.6	0.54	1.4	1.41	2.99	0.56	1.16
$P_{6,5}$	0	6.6	0.02	4.47	1.63	6.8	0	0.01
$P_{6,10}$	0	7.81	0.9	2.94	0.9	1.01	0.82	3.01
$P_{7,6}$	0	1.44	3.52	4.72	1.26	2.71	0	1.34
$P_{7,5}$	2.35	30	0.24	16.76	2.29	9.57	0	0
$P_{7,10}$	0	3.04	0.01	1.66	0	1.65	0	1.97

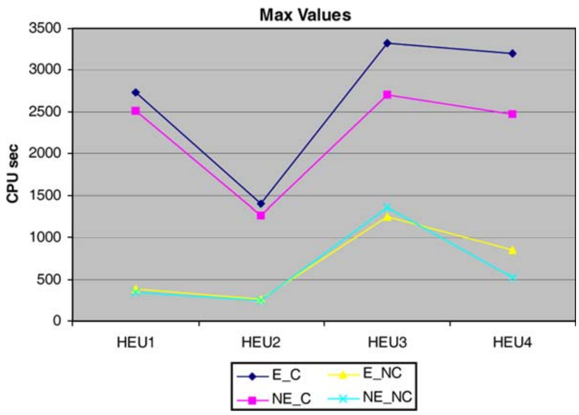

Fig. 1. Maximum CPU times.

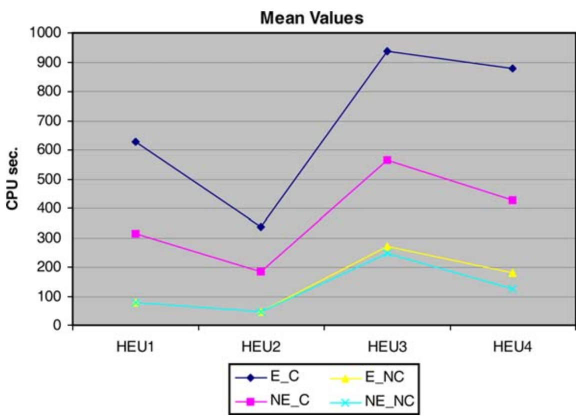

Fig. 2. Mean CPU times.

sistent scenario trees, the efficiency is almost the same in the equal and unequal due-dates case. Finally, in case of consistent scenarios tree, equal due dates required a much larger computation time.

5. Conclusions

In this paper we have proposed a multi-stage stochastic mixed-integer programming formulation of the lot-sizing and scheduling problem with uncertain processing times. In order to generate sub-optimal solutions, we have designed efficient heuristic strategies, based on the idea of decomposing the original problem into a sequence of smaller subproblems. Computational results on a large set of test problems have shown that our approach is able to provide high quality feasible solutions within a short amount of time.

References

Ahmed, S., Sahinidis, N., 2003. An approximation scheme for stochastic integer programs arising in capacity expansion. Operations Res. 51, 461–471.

Ahmed, S., King, A.J., Parija, G., 2003. A multistage integer programming approaches for capacity expansion under uncertainty. J. Global Optimization 26, 3–24.

Allahverdi, A., Gupta, J.N.D., Aldowaisan, T., 1999. A review of scheduling research involving set-up consideration. Omega 27, 219–239.

Alonso, A., Escudero, L., Ortuno, M.T., 2000. A stochastic 0-1 program based approach for the air traffic flow management problem. Europ. J. Operational Res. 120, 47–62.

Beraldi, P., Ghiani, G., Grieco, A., Guerriero, E. Heuristic algorithms for the scheduling of parallel machines with sequence dependent set-up costs. J. Scheduling, submitted for publication.

Clark, A.R., Clark, S.J., 2000. Rolling-horizon lot-sizing when setup times are sequence dependent. Int. J. Product. Res. 38, 2287–2308.

Dearing, P.M., Henderson, R.A., 1984. Assigning looms in a textile weaving operation with changeover limitation. Product. Inventory Manage. 25, 23–31.

Dillenberger, Ch., Escudero, L.F., Wollensak, A., Zhang, W., 1994. On practical resource allocation for production planning and scheduling with period overlapping setups. Europ. J. Operational Res. 75, 275–286.

Drexel, A., Kimms, A., 1997. Lot sizing and scheduling; survey and extensions. Europ. J. Operational Res. 99, 221–235.

Escudero, L.F., Salmeron, J. On a fix-and-relax framework for large-scale resource-constrained project scheduling. Ann. Operations Res., to appear.

Guerriero, E., 2003. Exact and heuristic algorithms for lot sizing and scheduling problems. Ph.D. Thesis, Universitá della Calabria, Rende (CS), Italy.

Haen, K., Løkketangen, A., Woodruff, D.L., 2001. Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. Europ. J. Operational Res. 132, 116–122.

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

Lulli, G., Sen, S., 2004. A branch-and-price algorithm for multi-stage stochastic integer programming with application to stochastic batch-sizing problems. Manage. Sci. 50 (6), 786–796.

Meyer, H., 2002. Simultaneous lot-sizing and scheduling on parallel machines. Europ. J. Operational Res. 139, 277–292.

Pinedo, M., 1995. Scheduling Theory, Algorithms and Systems. Prentice-Hall, Englewood cliffs, NJ.

Pochet, Y., 2001. Mathematical programming models and formulations for deterministic production planning problems. In: Jünger, M., Naddef, D. (Eds.), Computational Combinatorial Optimization. Lecture Notes in Computer Science. Springer, Berlin, pp. 57–111.

Sen, S., 2003. Algorithms for Stochastic Mixed-Integer Programming Models. Working paper, MORE Institute, SIE Dept., University of Arizona, Tucson, AZ.

Sox, C.R., Jackson, P.L., Bowman, A., Muckstadt, J.A., 1999. A review of the stochastic lot scheduling problem. Int. J. Product. Econ. 62, 181–200.

Staggemeier, A.T., Clark, A.R., 2001. A survey of lot-sizing and scheduling models. Proceedings of the 23rd Annual Symposium of the Brazilia Operational Research Society, Campos do Jordo, Brazil.

Staggemeier, A.T., Clarkand, A.R., Aickelin, U., Smith, J., 2002. A hybrid genetic algorithm to solve a lot-sizing and scheduling problem. Report MS-2002-4, Intelligent Computer Systems Centre Group, University of the West of England, Bristol, Great Britain.

http://mally.eco.rug.nl/spbib.html.

Wagner, H.M., Whitin, T.M., 1958. Dynamic version of the economic lot size model. Manag. Sci. 5, 89–96.

Wolsey, L.A., 1997. MIP modelling of changeovers in production planning and scheduling problems. Europ. J. Operational Res. 99, 154–161.