Computers & Operations Research 39 (2012) 1977–1987

Contents lists available at SciVerse ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

A stochastic production planning problem with nonlinear cost

Lixin Tang [a,*], Ping Che [a,b], Jiyin Liu [a,c]

[a] Liaoning Key Laboratory of Manufacturing System and Logistics, The Logistics Institute, Northeastern University, 110004 Shenyang, China
[b] The Department of Mathematics, Northeastern University, 110004 Shenyang, China
[c] Business School, Loughborough University, Leicestershire LE11 3TU, UK

ARTICLE INFO

Available online 12 September 2011

Keywords:
Stochastic production planning
Inventory
Scenario-based approach
Stepwise Lagrangian relaxation
Variable splitting
MINLP

ABSTRACT

Most production planning models are deterministic and often assume a linear relation between production volume and production cost. In this paper, we investigate a production planning problem in a steel production process considering the energy consumption cost which is a nonlinear function of the production quantity. Due to the uncertain environment, the production demands are stochastic. Taking a scenario-based approach to express the stochastic demands according to the knowledge of planners on the demand distributions, we formulate the stochastic production planning problem as a mixed integer nonlinear programming (MINLP) model.

Approximated with the piecewise linear functions, the MINLP model is transformed into a mixed integer linear programming model. The approximation error can be improved by adjusting the linearization ranges repeatedly. Based on the piecewise linearization, a stepwise Lagrangian relaxation (SLR) heuristic for the problem is proposed where variable splitting is introduced during Lagrangian relaxation (LR). After decomposition, one subproblem is solved by linear programming and the other is solved by an effective polynomial time algorithm. The SLR heuristic is tested on a large set of problem instances and the results show that the algorithm generates solutions very close to optimums in an acceptable time. The impact of demand uncertainty on the solution is studied by a computational discussion on scenario generation.

1. Introduction

Iron and steel industry is an essential sector in economy. This sector consumes extensive energy since most iron and steel production operations are performed at high temperature. Energy cost accounts for a large proportion of the total production cost. Therefore, energy saving in production is of great significance for steel companies to reduce cost and stay competitive. This motivates us to study the production planning problem in a hot rolling mill with the objective to minimize the energy consumption.

In the hot rolling production process studied in this paper, steel slabs are first heated up to the required temperature in the heat furnace. Then the heated slabs are rolled, on the hot rolling mills, into hot strips according to the specification of the demands. Finally, the hot strips are temporarily stored in the strip yard waiting to be delivered to customers or to be further processed in the downstream processing stages. This integrated production process is illustrated in Fig. 1. For convenience, we will call the slabs or strips products hereafter. In practice there are capacity restrictions for heating, hot rolling as well as stock holding. The total cost of the integrated production process includes production cost, especially the cost on energy consumption, and inventory cost for finished products. For each type of product produced in a period, the cost on rolling includes a setup cost associated with the product type and a variable cost proportional to the production quantity. Due to the special feature of the heat furnace, its energy consumption cost is nonlinear with respect to the production quantity. So the production planning in this integrated steel production process needs to consider nonlinear cost.

The production process is affected by the demands of products which are full of uncertainty. The uncertainty may be caused by production plan changes in downstream stages or the random arrival of customer orders. Ultimately and to a large extent it comes from fluctuations in the world steel market. For example, demands of steel products are affected by the prices which are in turn affected by the prices of energy and raw materials such as iron ore. On the other hand, demands also have an impact on prices. Moreover, demands of steel products are also influenced by the economic situation, government policies, protectionism, etc. Because of the variety of direct and indirect impacting factors, demands can hardly be forecasted accurately. Therefore this problem needs to consider stochastic demands caused by these factors.

There has been extensive research in the general field of production planning with a large proportion concerning linear production cost. In terms of deterministic circumstances, Bahl et al. [2] provide a survey about lot sizing problems in production planning and Karimi

* Corresponding author. Tel./fax: +86 24 83680169.
E-mail addresses: qhjytlx@mail.neu.edu.cn (L. Tang), cheping@mail.neu.edu.cn (P. Che), j.y.liu@lboro.ac.uk (J. Liu).

0305-0548/$ - see front matter © 2011 Published by Elsevier Ltd.
doi:10.1016/j.cor.2011.09.007

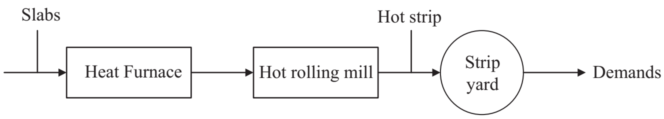

Fig. 1. The production process under consideration.

et al. [13] provide a review of models and algorithms for single-level multi-product capacitated lot sizing problems. In addition, Xie and Dong [24] propose heuristic genetic algorithms for general capacitated lot sizing problems. Jans and Degraeve [11] review various meta-heuristics that are specially developed for lot sizing problems. Minner [14] analyzes three simple heuristics for multi-product dynamic lot sizing problem with limited warehouse capacity. In terms of stochastic circumstances, a review of models for production planning under uncertainty is given by Mula et al. [15]. Sox [19] develops an optimal solution algorithm for the single-item dynamic lot sizing problem with random demand and non-stationary costs. Bakir and Byrne [3] develop a two-stage stochastic linear programming model for the multi-product multi-period problem with stochastic demands. Haugen et al. [10] address a stochastic version of the classical W-W model (Wagner and Whitin [23]) with an extension of the backlogging possibility and develop a meta-heuristic based on progressive hedging. The model they study is single-item and no capacity constraints are considered. In recent years, Brandimarte [5] considers a stochastic version of the classical multi-item capacitated lot sizing problem and propose a heuristic algorithm based on a fix-and-relax strategy. Azaron et al. [1] develop a polynomial algorithm for the single-product multi-period lot sizing problem with concave inventory cost and linear stochastic production cost. Related to steel industry, Tang et al. [21] provide a review of planning and scheduling methods for integrated steel production. They classify the optimization methods for steel production planning and scheduling into four types and review literature associated with each type.

Production planning problems concerning nonlinear production cost are not so widely investigated as that of linear production cost. Gutiérrez et al. [9] address a single-item lot sizing problem with uncertainty demands and concave production and holding costs. Rizk et al. [17] investigate the multi-item lot sizing problems with piecewise linear resource costs. Chazal et al. [7] study the deterministic single-product production planning problem of a profit-maximizing firm with convex production and storage cost. Since the production planning problem we are facing includes multiple items with stochastic demands and nonlinear production cost, previous approaches are not applicable.

In the field of electricity generation, the generation cost is convex increasing [20], which has a similar feature to the cost in our problem. However, there are many differences between the electricity generation problem and the production planning problem in this paper. A main difference is that electricity is non-storable so that no inventory cost needs to be considered. Therefore, approaches developed for the electricity generation problem cannot be applied to our problem.

In this paper, the studied multi-item multi-period production planning problem with nonlinear production cost and stochastic demands is formulated as a MINLP model according to a scenario-based approach. The objective is to minimize both the inventory cost and the production cost. As the problem is large-scale and it is impractical to solve it by using a commercial solver, a heuristic algorithm is devised for this problem. The main difficulty in developing the heuristic is in dealing with the nonlinear cost in the objective function. Oh and Karimi [16] take a multi-segment separable programming approach to help solving the nonlinear difficulty met in their lot-sizing problem. Taking a similar approach, we propose a SLR heuristic for our problem. The proposed heuristic is tested and its effectiveness is verified through computational experiments.

The remainder of the paper is organized as follows. In the next section, the MINLP model is formulated for the problem after the energy consumption feature of the heat furnace is discussed in detail. Section 3 is devoted to devising our heuristic for the problem. An approximate formulation is developed and a SLR heuristic is presented. In Section 4, computational experiments are reported. Finally, the paper is ended with our conclusions and directions for future research in Section 5.

2. Problem description and formulation

We consider the production planning problem for the integrated production process from the heat furnace, through hot rolling, to the storage yard of rolled products. The objective is to minimize the total production and storage cost. The production cost in hot rolling and the storage cost of rolled products are typical linear functions of production quantity and inventory level, respectively, while the cost of heat furnace is mainly on energy consumption. The production plan needs to consider stochastic demands for products and backlogging is not permitted. Based on the practical production process, there is no delay between the heating process and hot rolling because heated products are charged directly into the hot rolling mill as soon as they come out of the heat furnace. The production planning problem under consideration takes a day as its planning period.

The rest of this section is organized as follows. The special feature of the energy consumption cost of the heat furnace is discussed in Section 2.1. The way of modeling stochastic demands is discussed in Section 2.2. A MINLP model for the stochastic production planning problem is formulated in Section 2.3.

2.1. Energy consumption feature of the heat furnace

There are different types of heat furnaces in the steel-rolling mill. Each type of furnace has a different relation between the energy consumption and the products quantity heated. Here, the furnace considered is of the type for which the relative distribution of heat along the furnace length is fixed. It does not change with the products quantity or the heat provided in the furnace. Meanwhile, the state of the heat furnace usually remains “on” since starting up the furnace needs a considerable cost. When there is no product in the furnace, a fixed energy consumption cost is incurred for heat preservation of the furnace. According to the work feature of this type of heat furnace, Yu et al. [25] develop the relationship between the quantity of the products heated and the energy consumption cost per period as

$$E = E_0 \exp(rx), \tag{1}$$

Here E denotes the energy consumption cost and x denotes the amount of products heated in the heat furnace per period. E_0 is a constant indicating the energy consumption cost when there is no product in the heat furnace ($x=0$) and r is a positive constant associated with the furnace. Equivalently, the energy consumption cost per unit of production quantity per period is

$$E/x = E_0 \exp(rx)/x. \tag{2}$$

It can be seen that function E/x is convex and achieves its minimum value at $x_0 = 1/r$. Clearly if only the energy consumption cost in the furnace is considered, $x_0 = 1/r$ is the optimal production quantity.

The above analysis indicates that there is an exponential relationship between the heating quantity and the unit energy consumption in the heat furnace. Therefore it is not realistic to assume the production cost as a linear function of the production quantity in the system. To model the production planning problem accurately, we adopt the nonlinear unit energy consumption cost function (2) to express the associated production cost.

2.2. Modeling the stochastic demands

Statistical method for dealing with uncertainty is to express an uncertain parameter as a random variable with its distribution function. However, obtaining the distribution function exactly is not easy usually because of lacking enough information. Even if the exact distribution is given, the resulting stochastic optimization problem is notoriously hard to solve. Here, we use a scenario-based approach to express the uncertain demands as a set of demand scenarios according to the knowledge of the planners on the demand distributions. This is a popular approach for modeling uncertainty and has been effectively used in the stochastic unit commitment problem [20], the stochastic technology choosing and capacity planning problem [8] and many other stochastic problems.

According to this approach, the possible evolution cases of the stochastic demands throughout the planning horizon are assumed to be of finite number and represented with a set of demand scenarios. Meanwhile, each scenario is assigned a weight to reflect the probability of its realization. That is, we describe the stochastic demands as follows:

$$P\{\tilde{d}_{it} = d_{it}^s,\ i \in I,\ t \in T\} = P_s, s \in S,$$

where S is the set of demand scenarios, I is the set of products, T is the set of periods in the planning horizon, d_{it} is the stochastic demand for product i in period t, $i\in I$, $t\in T$, and d_{it}^s is the demand for product i in period t in scenario s, $i\in I$, $t\in T$, $s\in S$, and P_s is the probability of scenario s, $s\in S$. To describe the evolution feature of the stochastic demands, the scenarios are modeled as a tree structure and a scenario is represented as a path from the root node to a terminal node as illustrated in Fig. 2. For a given period t, the demand realization in the subsequent periods is invisible and non-anticipatable. The branches in the subsequent periods may express the possible realization.

Fig. 2 shows that some scenarios may share the same demand path from the first period up to a certain time period. Scenarios 1, 2, and 3 share the same demand path in period 1 and $d_{11}^1 = d_{11}^2 = d_{11}^3 = 50$, $d_{21}^1 = d_{21}^2 = d_{21}^3 = 50$. Scenarios 2 and 3 share the same demand path from period 1 to period 2 and $d_{12}^2 = d_{12}^3 = 52$, $d_{22}^2 = d_{22}^3 = 53$. Because of the invisibility and non-anticipativity feature in the subsequent periods, scenarios sharing the same demand path before and in period t are indistinguishable in period t. In Fig. 2, scenarios 1, 2, and 3 are indistinguishable in period 1 and

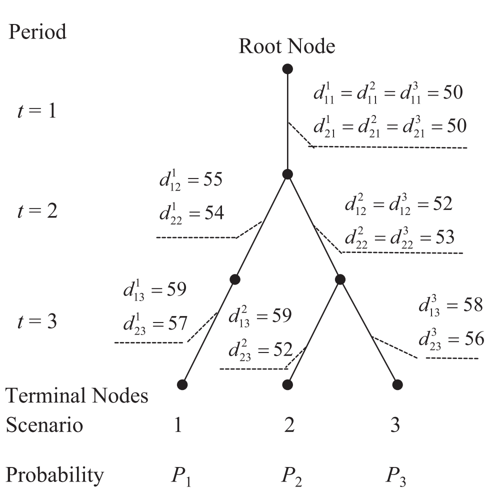

Fig. 2. An example of scenario tree for stochastic demands of two products over three periods.

scenarios 2 and 3 are indistinguishable in period 2. For the indistinguishability of scenario information, we use the term, scenario bundle, mentioned by Rockafellar and Wets [18] to formulate it. A scenario bundle in a period is a set of indistinguishable scenarios in this period. Two scenarios, s and j, are members of the same bundle in period t if and only if $d_{i\tau}^j = d_{i\tau}^s$ holds for $1 \le \tau \le t$ and all i. Obviously, each scenario belongs to only one bundle in a period. Let $Q(t, s)$ denote the set of scenarios corresponding to the scenario bundle including scenario s in period t. Then in Fig. 2, $Q(1, 1)=Q(1, 2)=Q(1, 3)=\{1, 2, 3\}$, $Q(2, 1)=\{1\}$, $Q(2, 2)=Q(2, 3)=\{2, 3\}$, $Q(3, 1)=\{1\}$, $Q(3, 2)=\{2\}$, $Q(3, 3)=\{3\}$.

If two scenarios are indistinguishable in period t, the associated decisions made for these scenarios in period t are the same. In Fig. 2, the decisions made for scenario 2 in period 2 are the same as those for scenario 3. Based on the scenario bundle defined above, this decision feature will be reflected by introducing the indistinguishability constraints to the formulation of the problem.

2.3. The model

We first define the parameters and decision variables in the following:

Parameters

- c_{it} unit hot rolling cost for product i in period t, $i\in I$, $t\in T$;
- h_{it} unit inventory holding cost for product i in period t after hot rolling, $i\in I$, $t\in T$;
- Se_{it} setup cost for product i in period t, $i\in I$, $t\in T$;
- $Inv^{\max}$ capacity for inventory holding of the store yard after hot rolling;
- Inv_{i0} initial inventory of product i over the planning horizon, $i\in I$;
- $HF_t^{\max}$ maximum amount of products allowed to be heated in the heat furnace in period t, $t\in T$;
- $HR_t^{\max}$ maximum amount of products allowed to be rolled on the hot rolling mills in period t, $t\in T$;
- α yield ratio from slabs to strips, i.e., the amount of strips produced by one unit amount of slabs; Its value is positive and smaller than 1;
- M a positive real number large enough.

Decision variables

- x_{it}^s amount of product i entering the heat furnace and hot rolling mills in period t in scenario s, $i\in I$, $t\in T$, $s\in S$;
- y_{it}^s amount of product i produced from hot rolling mills in period t in scenario s, $i\in I$, $t\in T$, $s\in S$;
- z_{it}^s a binary variable indicating whether product i is rolled on the hot rolling mill in period t in scenario s, $i\in I$, $t\in T$, $s\in S$;
- Inv_{it}^s inventory of product i at the end of period t in scenario s, $i\in I$, $t\in T$, $s\in S$.

In order to distinguish between the problem and its approximate version appeared in the later section, we call the problem after modeling the stochastic demands using the scenario-based approach the original problem which can be formulated as follows:

Original problem (OP)
Minimize g_O, with

$$g_O = \sum_{s\in S} P_s \left[E_0 \sum_{t\in T} \exp\left(r \sum_{i\in I} x_{it}^s \right) + \sum_{i\in I}\sum_{t\in T} (c_{it}x_{it}^s + h_{it}Inv_{it}^s + Se_{it}z_{it}^s) \right] \quad (3)$$

subject to

$$Inv^s_{i,t-1}+y^s_{it}=d^s_{it}+Inv^s_{it},\ \ i\in I,\ \ t\in T,\ \ s\in S \tag{4}$$

$$\alpha x^s_{it}=y^s_{it},\ \ i\in I,\ \ t\in T,\ \ s\in S \tag{5}$$

$$\sum_{i\in I} Inv^s_{it}\le Inv^{\max},\ \ t\in T,\ \ s\in S \tag{6}$$

$$\sum_{i\in I} x^s_{it}\le HF^{\max}_t,\ \ t\in T,\ \ s\in S \tag{7}$$

$$\sum_{i\in I} x^s_{it}\le HR^{\max}_t,\ \ t\in T,\ \ s\in S \tag{8}$$

$$x^s_{it}\le Mz^s_{it},\ \ i\in I,\ t\in T,\ \ s\in S \tag{9}$$

$$x^j_{it}=x^s_{it},\ \ j\in Q(t,\,s),\ i\in I,\ \ t\in T,\ \ s\in S \tag{10}$$

$$y^j_{it}=y^s_{it},\ \ j\in Q(t,\,s),\ i\in I,\ \ t\in T,\ \ s\in S \tag{11}$$

$$z^j_{it}=z^s_{it},\ \ j\in Q(t,\,s),\ \ i\in I,\ \ t\in T,\ \ s\in S \tag{12}$$

$$Inv^j_{it}=Inv^s_{it},\ \ j\in Q(t,\,s),\ i\in I,\ t\in T,\ s\in S \tag{13}$$

$$x^s_{it}\ge 0,\ y^s_{it}\ge 0,\ Inv^s_{it}\ge 0,\ \ i\in I,\ \ t\in T,\ \ s\in S \tag{14}$$

$$z^s_{it}:0-1,\ \ i\in I,\ \ t\in T,\ \ s\in S \tag{15}$$

Expression (3) offers the optimization objective. The objective is to minimize the weighted average costs under all scenarios over the planning horizon for heating, hot rolling, and inventory holding. Each weight of the cost is the probability of the associated scenario. Eqs. (4) represent the inventory balance constraints. Eqs. (5) reflect the fact that there is an output-to-input ratio for hot rolling due to the material losses in the process. Inequalities (6) represent capacity constraints of inventory holding. Inequalities (7) and (8) represent production capacity constraints for heating and hot rolling, respectively. Inequalities (9) show the variables consistency between x^s_{it} and z^s_{it}. Eqs. (10)–(13) indicate the indistinguishability constraints on decision variables. Inequalities (14) define the nonnegative value fields for the continuous variables, while constrains (15) show the binary evaluation of the integer variables.

Since g_O increases as the production quantity increases and no backlogging is permitted, the final inventories in the optimal solution must be zero, i.e.,

$$Inv^s_{i|T|}=0,\ \ i\in I,\ \ s\in S \tag{16}$$

where $|\cdot|$ is the norm of a set. Based on constraints (4), (5), and (16), we have

$$Inv^s_{it}=Inv_{i0}+\sum_{k\in T,k\le t}(\alpha x^s_{ik}-d^s_{ik}). \tag{17}$$

Replacing y^s_{it} and Inv^s_{it} in (3) with (5) and (17), respectively, we get

$$g_O=\sum_{s\in S}P_s\left\{E_0\sum_{t\in T}\exp\left(r\sum_{i\in I}x^s_{it}\right)+\sum_{i\in I}\sum_{t\in T}\left[\left(c_{it}+\alpha\sum_{k\in T,k\ge t}h_{ik}\right)x^s_{it}+Se_{it}z^s_{it}\right]\right\} \tag{18}$$

Here a constant term, $F=\sum_{i\in I}\sum_{t\in T}Inv_{i0}h_{it}-\sum_{s\in S}P_s[\sum_{i\in I}\sum_{t\in T}(\sum_{k\in T,k\ge t}h_{ik})d^s_{it}]$, is omitted from (18) since it has no effect on the solution.

Compared with the traditional production planning model, the classical W-W model for example, the above model shows the following characteristics.

(1) The W-W model considers the single-item problem and the main task is to minimize only the sum of two types of costs, order cost and stock holding cost. Our model includes multiple items, which are interconnected through the capacity constraints. Besides, our model considers the balance of more cost elements including setup cost, production cost proportional to production quantity, energy consumption cost, and inventory holding cost.
(2) Since the energy consumption cost includes an exponential function, the objective function is nonlinear, which results in a MINLP problem and endows the problem solving with challenge.
(3) Demands are stochastic, which is closer to the practical production. Meanwhile, the approach expressing uncertainty introduces a set of scenarios. This adds another dimension, scenario, to the model and makes it different from deterministic models. However, this approach inevitably results in a large-scale MINLP problem since the problem size increases as the number of scenarios associated with the uncertainty increases. For example, in a problem with $|T|$ periods and a scenario tree structure illustrated in Fig. 3, the number of scenarios is $2^{|T|-1}$, which increases exponentially as the planning horizon extends.

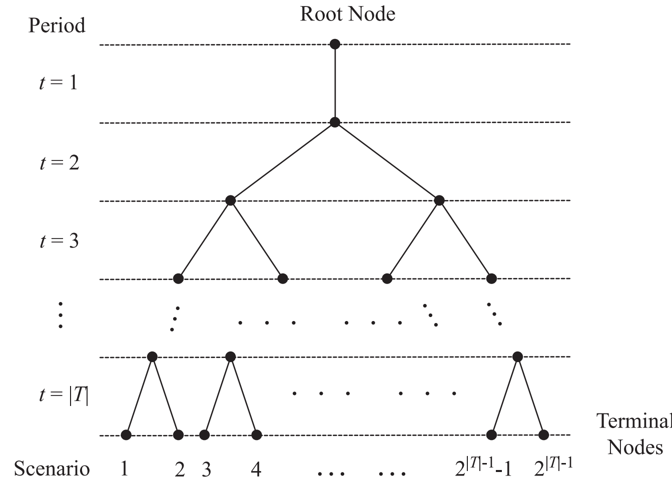

Fig. 3. An example of scenario tree structure with a $|T|$-period horizon.

3. Heuristic algorithm

In this part, a heuristic is devised for problem OP. It is a stepwise procedure alternately carrying out the linear approximation and applying the LR solution algorithm. In each cycle, the procedure includes two stages. In the first stage, the approximation range is updated according to the results of the last cycle so that a better solution may be found and the exponential term in problem OP is approximated linearly over this range. In the second stage, the approximate problem is solved using a variable splitting-based LR algorithm. Based on our heuristic, an upper bound for the optimal objective values of both the original problem and the approximate problem is presented. At the end of this part, a comparison between this paper and that of Carøe and Schultz [6] is given.

3.1. Linear approximation

To handle the nonlinear intractability of the problem, the linear approximation is adopted to over-approximate the nonlinear term $E_0\exp(r\sum_{i\in I}x^s_{it})$ with a piecewise linear function. First, we impose a lower bound $x^{sL}_{it}\ge 0$ and an upper bound x^{sU}_{it} on x^s_{it} since x^s_{it} is nonnegative and finite. The lower and upper bounds can be found by analyzing the constraints on x^s_{it}. A pair of obvious bounds is $x^{sL}_{it}=0$ and $x^{sU}_{it}=H^{\max}_t$, where $H^{\max}_t=\min\{HF^{\max}_t,\,HR^{\max}_t\}$. Then, the interval $[\sum_{i\in I}x^{sL}_{it},\,\sum_{i\in I}x^{sU}_{it}]$ is divided into $|H|$ equal subintervals with the length of $\Delta^s_t=\sum_{i\in I}\Delta^s_{it}$, where H is the set of subintervals indexed with h and $\Delta^s_{it}=(x^{sU}_{it}-x^{sL}_{it})/|H|$. Finally, over-approximating the curve

$E_0 \exp(r\sum_{i\in I} x_{it}^s)$ with a piecewise linear function, we get

$$E_0 \exp\left(r\sum_{i\in I} x_{it}^s\right) \approx E_0 \exp\left(r\sum_{i\in I} x_{it}^{sL}\right) + \sum_{i\in I}\sum_{h\in H} slope_t^{sh} x_{it}^{sh} \tag{19}$$

where

$$x_{it}^{sh} \geq 0, \quad i\in I, \quad t\in T, \quad s\in S, \quad h\in H \tag{20}$$

$$x_{it}^s = x_{it}^{sL} + \sum_{h\in H} x_{it}^{sh}, \quad i\in I, \quad t\in T, \quad s\in S, \tag{21}$$

$$slope_t^{sh} = E_0 \exp\left(r\sum_{i\in I} x_{it}^{sL}\right)[\exp(rh\Delta_t^s) - \exp(r(h-1)\Delta_t^s)]/\Delta_t^s$$

is the slope of the piecewise linear function in the hth subinterval. Furthermore, to make the approximation mathematically legitimate, the following conditions must be satisfied:

$$x_{it}^{sh} \leq \Delta_{it}^s, \quad i\in I, \quad t\in T, \quad s\in S, \quad h\in H \tag{22}$$

$$x_{it}^{s,h+1} = \cdots = x_{it}^{s|H|} = 0 \;\; if \;\; x_{it}^{sh} = 0, \quad i\in I, \; t\in T, \; s\in S, \; h\in H \tag{23}$$

$$x_{it}^{s1} = \cdots = x_{it}^{s,h-1} = \Delta_t^s \;\; if \;\; x_{it}^{sh} \neq 0, \quad i\in I, \; t\in T, \; s\in S, \; h\in H. \tag{24}$$

Replacing the exponential function in (18) with (19), we can obtain the approximate formulation of problem OP as follows.

Approximate problem (AP)
Minimize g_A, with

$$g_A = \sum_{s\in S} P_s\left[\sum_{i\in I}\sum_{t\in T}\sum_{h\in H}\left(slope_t^{sh} + c_{it} + \alpha\sum_{k\in T, k\geq t} h_{ik}\right)x_{it}^{sh} + \sum_{i\in I}\sum_{t\in T} Se_{it} z_{it}^s\right] \tag{25}$$

subject to

$$Inv_{i0} + \alpha\sum_{k\in T, k\leq t} x_{ik}^s \geq \sum_{k\in T, k\leq t} d_{ik}^s, \quad i\in I, t\in T, t\leq |T|-1, s\in S \tag{26}$$

$$Inv_{i0} + \alpha\sum_{t\in T} x_{it}^s = \sum_{t\in T} d_{it}^s, \quad i\in I, \; s\in S \tag{27}$$

$$\sum_{i\in I}\left[Inv_{i0} + \alpha\sum_{k\in T, k\leq t} x_{ik}^s - \sum_{k\in T, k\leq t} d_{ik}^s\right] \leq Inv^{\max}, \quad t\in T, \; s\in S \tag{28}$$

and constraints (7)–(10), (12), (15), (20)–(24).
In (25), a constant term

$$G = \sum_{s\in S} P_s\left[\sum_{t\in T} E_0 \exp\left(r\sum_{i\in I} x_{it}^{sL}\right) + \sum_{i\in I}\sum_{t\in T}\left(c_{it} + \alpha\sum_{k\geq t, k\in T} h_{ik}\right)x_{it}^{sL}\right],$$

is omitted since it has no effect on the solution.

Problem AP is NP-hard since one of its special cases, a multi-item lot-sizing problem with capacity constraints, has been proved to be NP-hard. Consequently, no polynomial time algorithm can be found to solve it exactly. For problem AP, we are going to devise a variable splitting-based LR algorithm according to the characteristics of the model.

3.2. Variable splitting-based LR

LR plays a primary role in dealing with large-scale separable mixed integer programming problems in the past decades. The key idea of this method is to relax the coupling constraints through introducing Lagrangian multipliers and decompose the complicated problem into some simple subproblems or many small-scale subproblems. Given the Lagrangian multipliers, the relaxed problem provides a lower bound for the optimal primal objective value in a minimization problem. Generally, by way of updating the Lagrangian multipliers effectively, the lower bound can be improved gradually.

To diminish the loss of information included in the coupling constraints and improve the performance of the algorithm, variable splitting is introduced into the LR algorithm. Variable splitting is an effective method for getting a stronger lower bound. Jörnsten and Näsberg [12] have used this approach in solving the generalized assignment problem. Barcia and Jörnsten [4] have improved this method by combining it with bound improving sequences. The main step of variable splitting is to transform the problem into an equivalent one through introducing artificial variables which are copies of some original decision variables. The resulting variable copy constraints are usually relaxed by the LR algorithm. By using this technique, problem AP can be solved as follows.

3.2.1. An equivalent problem

In problem AP, a set of artificial variables $\{az_{it}^s, \; i\in I, \; t\in T, \; s\in S : az_{it}^s \geq 0\}$ and the following variable copy constraints:

$$z_{it}^s = az_{it}^s, \; i\in I, \; t\in T, \; s\in S \tag{29}$$

are added and an equivalent problem is obtained below.

Equivalent problem (EP)
Minimize g_E, with

$$g_E = \sum_{s\in S} P_s\left[\sum_{i\in I}\sum_{t\in T}\sum_{h\in H}\left(slope_t^{sh} + c_{it} + \alpha\sum_{k\in T, k\geq t} h_{ik}\right)x_{it}^{sh} + \sum_{i\in I}\sum_{t\in T} Se_{it} az_{it}^s\right] \tag{30}$$

subject to

$$x_{it}^s \leq M az_{it}^s, \; i\in I, \; t\in T, \; s\in S \tag{31}$$

$$az_{it}^s \geq 0, \, i\in I, \; t\in T, \; s\in S \tag{32}$$

and constraints (7), (8), (10), (12), (15), (20)–(24), (26)–(29).

3.2.2. Relaxed problem

To give an explicit expression for the relaxed problem, we define $\pi(t,s)$ to be the scenario with the smallest index among the scenarios sharing the same bundle with scenario s in period t, i.e., $\pi(t,s) = \min\{j : j\in Q(t,s)\}$, $t\in T$, $s\in S$, and let

$$A_{tsj} = \begin{cases} 1, & \text{if } j = \pi(t,s), \\ 0, & \text{otherwise}, \end{cases} \quad t\in T, s\in S, j\in S.$$

Based on the above definitions, constraints (12) are expressed equivalently by

$$z_{it}^s = \sum_{j\in S} P_j z_{it}^j A_{tj\pi(t,s)} \Big/ \sum_{j\in S} P_j A_{tj\pi(t,s)}, \quad i\in I, \; t\in T, \; s\in S. \tag{33}$$

Relaxing coupling constraints (33) and (29) using Lagrangian multipliers $\mu_{1it}^s \in R$ and $\mu_{2it}^s \in R$, respectively, $i\in I$, $t\in T$, $s\in S$, we can generate the following relaxed problem.
(LRP)
Minimize $g_{LR}(\boldsymbol{\mu})$, with

$$g_{LR}(\boldsymbol{\mu}) = \sum_{s\in S} P_s\left[\sum_{i\in I}\sum_{t\in T}\sum_{h\in H}\left(slope_t^{sh} + c_{it} + \alpha\sum_{k\in T, k\geq t} h_{ik}\right)x_{it}^{sh} + \sum_{i\in I}\sum_{t\in T} Se_{it} az_{it}^s \right.$$
$$\left. + \sum_{i\in I}\sum_{t\in T}\mu_{1it}^s\left(z_{it}^s - \sum_{j\in S} P_j z_{it}^j A_{tj\pi(t,s)} / \sum_{j\in S} P_j A_{tj\pi(t,s)}\right) + \sum_{i\in I}\sum_{t\in T}\mu_{2it}^s (z_{it}^s - az_{it}^s)\right] \tag{34}$$

subject to constraints (7), (8), (10), (15), (20)–(24), (26)–(28), (31), and (32).
Here, $\boldsymbol{\mu}$ is the multiplier vector with elements μ_{qit}^s, $q=1,2$, $i\in I$, $t\in T$, $s\in S$. Problem LRP can be decomposed into two independent subproblems, LRP_1 and LRP_2, as follows.
($\text{LRP}_\mathbf{1}$)

Minimize $g_{LR1}(\boldsymbol{\mu})$, with

$$g_{LR1}(\boldsymbol{\mu}) = \sum_{s\in S} P_s \left[\sum_{i\in I}\sum_{t\in T}\sum_{h\in H} \left(slope_t^{sh} + c_{it} + \alpha \sum_{k\in T, k\geq t} h_{ik} \right) x_{it}^{sh} + \sum_{i\in I}\sum_{t\in T} (Se_{it} - \mu_{2it}^s) az_{it}^s \right] \tag{35}$$

subject to constraints (7), (8), (10), (20)–(22), (26)–(28), (31), and (32).

(LRP_2)

Minimize $g_{LR2}(\boldsymbol{\mu})$, with

$$g_{LR2}(\boldsymbol{\mu}) = \sum_{s\in S}\sum_{i\in I}\sum_{t\in T} P_s \left(\mu_{1it}^s + \mu_{2it}^s - \sum_{j\in S} \mu_{1it}^j P_j A_{ts\pi(t,j)} \Big/ \sum_{j\in S} P_j A_{ts\pi(t,j)} \right) z_{it}^s \tag{36}$$

subject to constraints (15).

Problem LRP_1 is a linear programming problem and can be solved optimally using standard optimization software. Since $\exp(r\sum_{i=1}^n x_{it})$ is convex, there must be $slope_t^{s1} < slope_t^{s2} < \cdots < slope_t^{sH}$. Therefore constraints (23) and (24) can be satisfied naturally by any solution to problem LRP_1 and are omitted from the problem. Since z_{it}^s is a binary variable, problem LRP_2 can be easily solved by using the following approach.

Step 1. Compute $B_{it}^s = \mu_{1it}^s + \mu_{2it}^s - \sum_{j\in S} \mu_{1it}^j P_j A_{ts\pi(t,j)} / \sum_{j\in S} P_j A_{ts\pi(t,j)}$, $i\in I$, $t\in T$, $s\in S$.
Step 2. Let $z_{it}^s = 1$ if $B_{it}^s < 0$ and $z_{it}^s = 0$ otherwise, $i\in I$, $t\in T$, $s\in S$.

3.2.3. Construction of a feasible solution

The solution to the relaxed problem is usually infeasible to problem EP due to the relaxation of the coupling constraints. Fortunately, the relaxation solution can be easily recovered to be feasible based on the optimal solution to problem LRP_1 by letting $az_{it}^s = z_{it}^s = 1$ if $x_{it}^s > 0$ and $az_{it}^s = z_{it}^s = 0$ otherwise. Taking the obtained feasible solution $X = \{x_{it}^s, az_{it}^s, z_{it}^s,\ i\in I,\ t\in T,\ s\in S\}$ as the initial solution, we adjust the decision variables according to the following strategy to improve the solution quality.

Step 1. Set $\bar{t} = T$.
Step 2. Set $\bar{s} = 1$ and $Vs = \emptyset$, where Vs is the set of visited scenarios in period $\bar{t}$.
Step 3. Adjust the production quantities between period $\bar{t}$ and period $\bar{t}-1$ without violating the indistinguishability constraints. If $x_{i\bar{t}}^{\bar{s}} > 0$ and $x_{i,\bar{t}-1}^{\bar{s}} > 0$, let

$$\bar{x}_{it}^s = \begin{cases} x_{i\bar{t}}^s - x_{i\bar{t}}^{\bar{s}}, & \text{if } t=\bar{t} \text{ and } s\in Q(\bar{t}-1, \bar{s}), \\ x_{it}^s, & \text{otherwise}, \end{cases}$$

$$\bar{x}_{i,t-1}^s = \begin{cases} x_{i,\bar{t}-1}^s + x_{i\bar{t}}^{\bar{s}}, & \text{if } t=\bar{t} \text{ and } s\in Q(\bar{t}-1, \bar{s}), \\ x_{i,t-1}^s, & \text{otherwise}, \end{cases}$$

$i\in I$, $t\in T$, $s\in S$, where $Q(t, s)$ is the scenario bundle that includes scenario s in period t, as defined in Section 2.2. If variables, $\bar{x}_{it}^s$, $i\in I$, $t\in T$, $s\in S$, are feasible to problem EP, let

$$\overline{az}_{it}^s = \bar{z}_{it}^s = \begin{cases} 1, & \text{if } \bar{x}_{it}^s > 0, \\ 0, & \text{otherwise}, \end{cases} \quad i\in I, t\in T, s\in S.$$

Step 4. If $\bar{X} = \{\bar{x}_{it}^s, \bar{z}_{it}^s, \overline{az}_{it}^s, i\in I, t\in T, s\in S\}$ can offer a lower upper bound for problem EP, update X with $\bar{X}$. Update Vs with $Vs \cup \{\bar{s}\}$.
Step 5. If $\bar{s} = |S|$, go to *Step* 6. Otherwise, set $\bar{s} = \bar{s}+1$.
Step 6. If there exists some scenario $j\in Vs$ satisfying $A_{\bar{t}\bar{s}j} = 1$, update Vs with $Vs \cup \{\bar{s}\}$ and go to *Step* 4. Otherwise, go to *Step* 3.
Step 7. If $\bar{t} = 2$, stop. Otherwise, set $\bar{t} = \bar{t}-1$ and go to *Step* 2.

The obtained feasible solution can provide an upper bound for problem EP.

3.2.4. Dual problem

To improve the lower bound, we update the Lagrangian multipliers by solving the following Lagrangian dual problem:

(LDP)

$$g_{LD} = \max_{\boldsymbol{\mu}\in R^{2|I||T||S|}} \min g_{LR}(\boldsymbol{\mu})$$

subject to the constraints in problem LRP, where $\min g_{LR}(\boldsymbol{\mu})$ is concave and piecewise linear continuous to $\boldsymbol{\mu}$. We use the subgradient method to optimize the dual problem by updating the multipliers at the mth iteration as follows:

$$\mu_{qit}^{s(m+1)} = \mu_{qit}^{s(m)} + s^{(m)} \xi_{qit}^{s(m)},\ q=1,2,\ i\in I,\ t\in T,\ s\in S,$$

where

$$s^{(m)} = \nu^{(m)}(UB - Z^L(\boldsymbol{\mu}^{(m)})) \Big/ \sum_{s\in S} P_s \left(\sum_{q=1,2}\sum_{i\in I}\sum_{t\in T} \xi_{qit}^{s(m)2} \right),$$

$$\xi_{1it}^{s(m)} = z_{it}^s(\boldsymbol{\mu}^{(m)}) - \sum_{j\in S} P_j z_{it}^j(\boldsymbol{\mu}^{(m)}) A_{tj\pi(t,s)} \Big/ \sum_{j\in S} P_j A_{tj\pi(t,s)},$$

$$\xi_{2it}^{s(m)} = z_{it}^s(\boldsymbol{\mu}^{(m)}) - az_{it}^s(\boldsymbol{\mu}^{(m)}),$$

$\nu^{(m)}$ is an iteration factor initialized by 2 and is halved when the lower bound remains unimproved for 2 consecutive iterations [22], UB is the minimum upper bound found so far and $Z^L(\boldsymbol{\mu}^{(m)})$ is the lower bound obtained at the mth iteration.

3.3. SLR heuristic

The obtained feasible solution to problem EP is also a feasible solution to problem OP since the two problems have the same constraints on variables x_{it}^s and z_{it}^s, $i\in I$, $t\in T$, $s\in S$. In order to decrease the error arising from the piecewise linear approximation of the nonlinear function with a finite $|H|$ and improve the quality of the solution to problem OP, we use a similar approach to that of Oh and Karimi [16] in the heuristic. The kernel of this approach is to contract repeatedly the interval over which the piecewise linear approximation is made with $|H|$ unchanged. It is a process of gradually zooming in on the original optimal solution. Based on the efforts above, the SLR heuristic is developed as follows:

Step 0. Initialization of the SLR heuristic: $u=0$; $x_{it(0)}^{sL} = 0$, $x_{it(0)}^{sU} = H_t^{\max}$, $i\in I$, $t\in T$, $s\in S$; $g_O^{SLR} = +\infty$, in which u is the cycle index, $x_{it(u)}^{sL}$ and $x_{it(u)}^{sU}$ are the upper bound and the lower bound of x_{it}^s, respectively for $i\in I$, $t\in T$, $s\in S$ in the uth cycle and g_O^{SLR} is the minimum objective value obtained so far for problem OP.
Step 1. Make linear approximation to problem OP as described in Section 3.1 and obtain problem AP.
Step 2. Implement the LR algorithm for problem AP as follows.
Step 2.0. Introduce the artificial variables into problem AP and obtain problem EP.
Step 2.1. Initialization of the LR algorithm. Set $m=0$, $UB=+\infty$, $LB=0$ and $\boldsymbol{\mu}^{(0)}=0$, where m is the iteration index, $\boldsymbol{\mu}^{(m)}$ is the Lagrangian multiplier at the mth iteration and LB is the maximum lower bound found so far for problem EP.
Step 2.2. Solving the subproblems. Given the Lagrangian multipliers, all the subproblems are optimally solved as described in Section 3.2.2. Both the optimal relaxation solution and the associated objective value, $g_{LR}^{(m)}$, for problem LRP can be get. If $g_{LR}^{(m)} > LB$, update LB with $g_{LR}^{(m)}$.
Step 2.3. Constructing a feasible solution. Based on the solution to problem LRP_1, a feasible solution to problem EP is constructed by calling the heuristic presented in Section 3.2.3 and is denoted by X. The corresponding objective values for problem EP and problem OP are denoted by $g_E^{(m)}$ and $g_O^{(m)}$, respectively. If $g_E^{(m)} < UB$, update UB with $g_E^{(m)}$. If $g_O^{(m)} < g_O^{SLR}$, update both g_O^{SLR} and X^* with $g_O^{(m)}$ and X,

respectively, where X^* is the best solution found so far to problem OP.

Step 2.4. If one of the following stopping criteria is met, we stop the iterations and go to *Step* 3.

(a) The relative duality gap$=(UB-LB)/LB<\varepsilon$, where ε is a small positive number.
(b) m is greater than the pre-specified iteration limit.
(c) The decrease of the gap remains less than η_1 for L_1 consecutive iterations.

Step 2.5. Update the Lagrangian multipliers as described in Section 3.2.4, set $m=m+1$, and go to *Step* 2.2.

Step 3. Updating the approximation intervals. The ranges over which the nonlinear cost is linearized are updated as follows:

$$x^{sL}_{it(u+1)}=\max\{x^{sL}_{it(0)},\ x^{s*}_{it}-\beta(x^{sU}_{it(u)}-x^{sL}_{it(u)})\},$$
$$x^{sU}_{it(u+1)}=\min\{x^{sU}_{it(0)},\ x^{s*}_{it}+\beta(x^{sU}_{it(u)}-x^{sL}_{it(u)})\},$$

where x^{s*}_{it} is the value of x^s_{it} corresponding to the best solution found so far to problem OP and $0<\beta<1$, $i\in I$, $t\in T$, $s\in S$.

Step 4. Judging if one of the cycle stopping criteria is satisfied. If one of the following criteria is satisfied, stop the procedure. Otherwise, set $u=u+1$ and go to *Step* 1.

(1) The length of the longest subinterval, $\max\{\Delta^s_{it}$, $i\in I, t\in T, s\in S\}$, is less than ς. Here ς is a small positive number.
(2) The improvement of g^{SLR}_O, $(g^{SLR}_O-g^{(m)}_O)/g^{(m)}_O$, remains less than η_2 for L_2 consecutive cycles.
(3) The maximum cycle number is reached.

In above procedure, η_q is a small positive number and L_q is a positive integer number designed by the user, $q=1, 2$.

Define UB^* as the minimum UB over all the cycles and we come to the following conclusions.

Proposition. *UB^* is the upper bound of the following costs:*

(i) *the optimal cost for problem OP;*
(ii) g^{SLR}_O, *the approximate optimal cost based on the SLR heuristic.*

Proof. (i) The conclusion is obvious because the linearization approximates the original objective function from above. (ii) Let X_0 denote the feasible solution corresponding to UB^* and $g_O(X_0)$ denote the objective value corresponding to X_0 in problem OP. Then we have $g_O(X_0)\le UB^*$ because of the over approximating. Because of the definition of g^{SLR}_O in *Step* 0 of the SLR heuristic, we have $g_O(X_0)\ge g^{SLR}_O$. Consequently, it comes to the conclusion. □

Define LB^* to be the LB in the cycle of finding UB^*. UB^* and LB^* will be used for evaluating the performance of our heuristic in the next section.

Carøe and Schultz [6] study the two-stage stochastic integer program problem with integer recourse and present an algorithm for it. In their paper, the second-stage variables are expressed by using the scenario-based approach to reflect the stochastic property, copies of the first-stage variables are introduced to make the objective function separable in scenario, and a LR algorithm is developed through relaxing the associated variable copy constraints which are exactly the indistinguishability constraints on the first-stage decision variables. There are several similar characteristics between our study and theirs. First, stochastic parameters and both continuous and integer decision variables exist in both models. Second, the scenario-based approach is used in both associated problem modeling. Third, variable splitting-based LR is used in both associated algorithm devising. However, since the structures of the two problems are obviously different, the solution methodologies are also different. First, the types of relaxed constraints are different. Instead of relaxing all the indistinguishability constraints on the variables, we relax the variable copy constraints and only the partial indistinguishability constraints which are associated with the integer variables. Second, the purposes of using variable splitting are different. It can be seen that our model, without introducing variable splitting, is of separable structure after the relaxation. Rather than to achieve a separable structure, we use variable splitting in LR to raise the lower bound. Third, we solve a mixed integer nonlinear model through progressively improving the linear approximation and using LR to deal with the linearized problem, while Carøe and Schultz [6] solve a mixed integer linear model using LR and branch-and-bound.

4. Experiments and numerical results

To test the performance of the SLR heuristic, numerical experiments are implemented as follows. First, the generation of problem instances is described in detail. Second, the values of the algorithm parameters used in the experiments are given. Third, computational results are reported. Finally, since the generation of demand scenario may affect the solution of the problem, a discussion on scenario generation is made according to the numerical results. The heuristic is coded in Visual C++ and run on a PC with 2.83 GHz and 3.25 GB memory. The linear programming involved is completed by using software package *CPLEX* 11.0.

4.1. Generation of problem instances

The problem instances are generated randomly on actual production data from the production process in Baoshan Iron and Steel Company, a most advanced steel company in China. Since the planning horizon of dynamic production plan is usually several days in actual production, we set it in the experiment as 3, 5, and 7 days. Based on the historical data, the number of products is set from 6 to 9. As we have no history information about the scenario tree when this work is done, the structure of the scenario tree is designed as illustrated in Fig. 3. In Fig. 3, we assume that there is one branch out of the root node since scenario trees with other branch structures at the root node can be expressed as a set of scenario trees with this branch structure. According to the production capacity and average product categories in a workday, demand is generated randomly from a uniform distribution, R_1: U(1000, 1200). For simplicity, demands in different periods in each scenario are generated independently. For convenience, this demand generation approach is denoted as D. Without loss of generality, all the scenarios are assumed to have equal probability of $1/|S|$.

Under each problem configuration, ten problem instances are generated randomly and their average results are taken for further analysis. Usually, the yield ratio in hot rolling process fluctuates in the range between 97.5% and 99.0% and we set it to 98% in all the instances. Since the average setup cost in hot rolling process is more than 50,000 and the largest one is far less than 100,000, setup cost in all the instances is generated randomly in the range from 50,000 to 100,000. To test the impact of different setup costs on the performance of the SLR heuristic, setup costs are generated from C_1: U(50,000, 65,000), or C_2: U(65,000, 80,000), or C_3: U(80,000, 100,000). The initial inventory is set to zero. The generation of the rest parameters is based on the historical data from the actual production as follows. The unit hot rolling cost is chosen randomly from U(250, 400). The unit stock holding cost is generated randomly from U(1, 10). The capacities for heating and hot rolling are generally the same and they are generated randomly from U(12,600, 18,500). Finally, the inventory holding capacity is generated randomly from U(30,000, 60,000).

4.2. Algorithm parameters

Some parameters in the SLR heuristic are configured as follows. In the SLR heuristic, a smaller ε, a smaller η_1 or a larger L_1 may bring more iterations and a better solution for problem AP. Similarly, a smaller ς, a smaller η_2 or a larger L_2 may bring more cycles and a better solution for problem OP. As a tradeoff between computational burden and accuracy, we let $\varepsilon=0.05$, $\varsigma=1$, $\eta_1=0.0002$, $\eta_2=0.00001$, $L_1=2$ and $L_2=2$. In addition, we let $|H|=3$ and $\beta=0.5$ in the experiment.

4.3. Performance of the LR algorithm for problem AP

Since problem AP is solved using the LR algorithm, the performance of the algorithm is tested in this subsection. As a core component of the SLR heuristic, the LR algorithm plays an important role because finding a good feasible solution for problem AP is beneficial for improving the quality of the solution to problem OP. Since a small gap in the LR algorithm indicates achieving a good feasible solution, we choose the relative duality gap of the algorithm as a measure. As the SLR heuristic includes multiple cycles of calling the LR algorithm, we sum up the relative duality gaps from all the cycles and obtain the gap mean through dividing the summation by the number of cycles.

The experimental results on the performance of the LR algorithm for problem AP are reported in columns 2–4 in Table 1, which show the average gap mean over ten instances under each problem configuration. For example, the average gap mean corresponding to the first configuration (problem size $6\times3\times4$ under the setup cost parameter structure C_1) in Table 1 is 0.0480, which is obtained by computing the gap mean over all the cycles for each of the ten instances and then averaging the gap means of the ten instances. From the left half of Table 1, some observations can be shown as follows. The total average gap mean over all problem configurations is 6.46%, which is acceptable and implicates the effectiveness of the LR algorithm. The gap mean increases as the setup cost or the problem size increases, which coincides with the characteristics of the relaxed constraints.

To further explore the quality of the upper bound generated by the LR algorithm for problem AP, we solve problem AP in each cycle by calling software package *CPLEX*. For each call, the time limit is 1000 s. Let g_A^{CPLEX} denote the best objective value achieved within the time limit. The relative deviation of the upper bound from g_A^{CPLEX}, $(UB-g_A^{CPLEX})/g_A^{CPLEX}$, in each cycle is computed and then the relative deviation mean over all the cycles is obtained. The associated computational results are listed in Table 2. In Table 2, columns 2–4 list the numbers of easy problem instances in which each problem AP can be optimally solved by using *CPLEX* within the time limit, denoted by NS_1, and columns 5–7 list the average relative deviation means over the easy instances. The average is not given if the number of easy instances is zero. It is observed that the average relative deviation means are small under all problem configurations and the total average is 0.56%. This implies that the feasible solutions generated by the LR algorithm are very close to the optimums of problem AP, which is beneficial to improve the performance of the SLR heuristic for the original problem. We also compute $(g_A^{CPLEX}-LB)/LB$, which is the relative deviation of g_A^{CPLEX} from the lower bound in each cycle. The corresponding average relative deviation means over ten problem instances are listed in columns 5–7 in Table 1. The computational results indicate that the varying trend of the deviation mean is agreeable to that of the gap mean and total average is 5.80%. From the whole results in Table 1, we can observe that the solution quality of the LR algorithm is very close to that of *CPLEX*.

The running times corresponding to Table 1 are listed in Table 3, in which unsolved problems contributing 1000 s in the average computation. Compared with the running time of the LR algorithm, the time consumed by *CPLEX* increases dramatically as the problem size increases. This is because problem AP is NP-hard and the running time of solving it by using software package is in exponential growth with the increasing of the problem size. This is why we solve problem AP by using the LR algorithm instead of *CPLEX*.

Table 1
Average relative duality gap means of the LR algorithm and average relative deviation means of g_A^{CPLEX} from LB for problem AP.

Size ($\lvert N\rvert\times\lvert T\rvert\times\lvert S\rvert$)	Gap mean			Deviation mean		
	C_1	C_2	C_3	C_1	C_2	C_3
$6\times3\times4$	0.0480	0.0599	0.0745	0.0444	0.0562	0.0683
$6\times5\times16$	0.0579	0.0735	0.0848	0.0510	0.0638	0.0733
$6\times7\times64$	0.0583	0.0722	0.0881	0.0500	0.0618	0.0740
$7\times3\times4$	0.0427	0.0527	0.0686	0.0405	0.0492	0.0635
$7\times5\times16$	0.0561	0.0686	0.0884	0.0506	0.0614	0.0772
$7\times7\times64$	0.0586	0.0747	0.0853	0.0516	0.0640	0.0718
$8\times3\times4$	0.0405	0.0525	0.0585	0.0391	0.0494	0.0545
$8\times5\times16$	0.0538	0.0670	0.0819	0.0492	0.0605	0.0731
$8\times7\times64$	0.0554	0.0723	0.0885	0.0494	0.0637	0.0767
$9\times3\times4$	0.0385	0.0393	0.0631	0.0372	0.0374	0.0604
$9\times5\times16$	0.0462	0.0676	0.0829	0.0427	0.0619	0.0747
$9\times7\times64$	0.0564	0.0631	0.0867	0.0515	0.0563	0.0764
Avg.	**0.0510**	**0.0636**	**0.0793**	**0.0464**	**0.0571**	**0.0703**
Total avg.			**0.0646**			**0.0580**

Table 2
Numbers of easy problem instances[a] and average relative deviation means of UB from g_A^{CPLEX}.

Size ($\lvert N\rvert\times\lvert T\rvert\times\lvert S\rvert$)	NS_1			Deviation mean		
	C_1	C_2	C_3	C_1	C_2	C_3
$6\times3\times4$	10	10	10	0.0034	0.0035	0.0058
$6\times5\times16$	10	10	10	0.0066	0.0091	0.0106
$6\times7\times64$	3	1	3	0.0078	0.0087	0.0140
$7\times3\times4$	10	10	10	0.0022	0.0034	0.0048
$7\times5\times16$	10	10	10	0.0052	0.0068	0.0103
$7\times7\times64$	0	0	0	–	–	–
$8\times3\times4$	10	10	10	0.0014	0.0030	0.0038
$8\times5\times16$	10	10	10	0.0043	0.0061	0.0081
$8\times7\times64$	0	0	0	–	–	–
$9\times3\times4$	10	10	10	0.0013	0.0018	0.0025
$9\times5\times16$	10	10	10	0.0034	0.0054	0.0076
$9\times7\times64$	0	0	0	–	–	–
Avg.	–	–	–	**0.0040**	**0.0053**	**0.0075**
Total avg.			–			**0.0056**

[a] Corresponding to the problem instances in which each problem AP can be optimally solved by using *CPLEX* within the time limit.

4.4. Performance of the SLR heuristic for problem OP

In this subsection, we test the performance of the SLR heuristic for the original problem. g_O^{SLR}, UB^*, LB^* and g_O^{opt} are chosen as the performance measures, where g_O^{SLR}, UB^* and LB^* are the ones defined in Section 3.3 and g_O^{opt} is the optimal objective value of problem OP. To obtain g_O^{opt}, we solve problem OP by using software package *LINGO* 9.0. Since the computational burden is extensive for obtaining an optimum for problem OP, the computational time in our experiment is limited to 1000 s. Taking into account that the optimums for some problem instances cannot be obtained within the given time limit, we also choose the best objective value generated by *LINGO* as a performance measure and denote it by g_O^{LINGO}. To give a clear comparison, the performance measures described above are normalized by LB^* and then reflected by the cost ratios. The computational results are reported in Tables 4 and 5, respectively. Table 4 reports the results on the easy problem instances that can be optimally solved using *LINGO* within the time limit while Table 5 reports the results on the solvable problem instances for which a feasible solution can be generated by using *LINGO* within the time limit. In Table 4, because none of the problem instances with $|T|=5$ or 7 can be optimally solved within the time limit, there are only the results for $|T|=3$. Under each problem configuration with $|T|=3$, the number of easy problem instances, denoted by NS_2, and the average cost ratios over the easy problem instances are listed. In Table 5, the numbers of solvable problem instances, denoted by NS_3, and the average cost ratios over the solvable problem instances are listed. Whenever no feasible solution can be generated for a problem instance, computation is abandoned for that instance. From Tables 4 and 5, we can make the following observations.

(1) Even when the problem size is small, not all the problem instances can be optimally solved using *LINGO*. When the problem size is medium or large, even the feasible solution cannot be generated by *LINGO* within the given time limit for some problem instances. Compared with *LINGO*, the SLR algorithm can solve all problem instances. These indicate that it is more practical to solve the problem by using the SLR algorithm than using *LINGO*.

(2) In Table 4, columns 5–7 list the average cost ratios of g_O^{opt} to LB^* under different problem configurations. Among the ratios, some are larger than 1 while others are smaller than 1. This is because LB^* is not a real lower bound of the original problem although it is a lower bound of the approximate problem. All the ratios in these columns are close to 1, which means LB^* is close to g_O^{opt}. Thus LB^* may be referred as a pseudo lower bound of the original problem to verify the performance of the heuristic when optimums are unavailable.

(3) Columns 8–10 in both Tables 4 and 5 list the average cost ratios of g_O^{SLR} to LB^* under different problem configurations. In Table 4, the total average value of g_O^{SLR}/LB^* is 0.98. This indicates g_O^{SLR} is close to LB^*, while LB^* is close to g_O^{opt} as mentioned above. It can be seen that the SLR heuristic has good performance on the small-scale problem instances, which is consistent with the result on a direct comparison between the SLR heuristic and *LINGO*. In Table 5, the total average value of g_O^{SLR}/LB^* is 1.02, which means g_O^{SLR} is close to

Table 3
Average running times of solving problem AP by using the LR algorithm and *CPLEX*, respectively (s).

Size ($|N|\times|T|\times|S|$)	LR			*CPLEX*		
	C_1	C_2	C_3	C_1	C_2	C_3
$6\times3\times4$	0.01	0.01	0.01	0.02	0.01	0.01
$6\times5\times16$	0.09	0.10	0.10	0.78	1.12	1.18
$6\times7\times64$	0.87	1.00	1.05	262.36	368.01	401.55
$7\times3\times4$	0.01	0.01	0.01	0.01	0.02	0.02
$7\times5\times16$	0.10	0.12	0.12	1.33	1.68	1.58
$7\times7\times64$	1.11	1.29	1.29	452.98	668.35	581.70
$8\times3\times4$	0.01	0.02	0.02	0.01	0.02	0.02
$8\times5\times16$	0.11	0.13	0.15	1.63	1.87	2.18
$8\times7\times64$	1.19	1.54	1.60	555.03	596.71	667.65
$9\times3\times4$	0.01	0.01	0.02	0.02	0.02	0.03
$9\times5\times16$	0.10	0.16	0.24	1.81	5.43	3.45
$9\times7\times64$	1.42	1.71	2.06	628.28	810.30	694.55
Avg.	**0.42**	**0.51**	**0.56**	**158.69**	**204.46**	**196.16**
Total Avg.			**0.49**			**186.44**

Table 4
Numbers of easy problem instances[a] and the average cost ratios.

Size ($|N|\times|T|\times|S|$)	NS_2			g_O^{opt}/LB^*			g_O^{SLR}/LB^*			UB^*/LB^*		
	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3
$6\times3\times4$	10	10	10	1.00	1.02	1.05	1.01	1.03	1.06	1.05	1.06	1.08
$7\times3\times4$	10	10	10	0.95	0.97	1.01	0.97	0.98	1.02	1.05	1.06	1.08
$8\times3\times4$	10	9	10	0.94	0.96	0.93	0.95	0.98	0.95	1.05	1.06	1.07
$9\times3\times4$	10	8	9	0.94	0.82	1.02	0.95	0.84	1.03	1.05	1.05	1.08
Total avg.			–			**0.97**			**0.98**			**1.06**

[a] Corresponding to the problem instances that can be optimally solved by using *LINGO* within the time limit.

Table 5
Numbers of solvable problem instances[a] and the average cost ratios.

Size ($|N|\times|T|\times|S|$)	NS_3			g_O^{LINGO}/LB^*			g_O^{SLR}/LB^*			UB^*/LB^*		
	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3	C_1	C_2	C_3
$6\times3\times4$	10	10	10	1.00	1.02	1.05	1.01	1.03	1.06	1.05	1.06	1.08
$6\times5\times16$	10	10	10	1.02	1.05	1.06	1.03	1.05	1.06	1.06	1.08	1.09
$6\times7\times64$	10	9	9	1.03	1.04	1.06	1.04	1.05	1.07	1.06	1.08	1.10
$7\times3\times4$	10	10	10	0.95	0.97	1.01	0.97	0.98	1.02	1.05	1.06	1.08
$7\times5\times16$	10	9	10	1.02	1.05	1.06	1.03	1.05	1.07	1.06	1.08	1.10
$7\times7\times64$	10	10	9	1.03	1.04	1.06	1.04	1.05	1.07	1.07	1.08	1.10
$8\times3\times4$	10	10	10	0.94	0.97	0.93	0.95	0.98	0.95	1.05	1.06	1.07
$8\times5\times16$	9	10	10	1.01	1.02	1.06	1.02	1.03	1.07	1.06	1.08	1.10
$8\times7\times64$	9	10	9	1.00	1.04	1.06	1.02	1.06	1.07	1.07	1.09	1.11
$9\times3\times4$	10	10	10	0.91	0.83	1.00	0.92	0.85	1.01	1.05	1.05	1.08
$9\times5\times16$	9	10	10	0.90	1.04	1.06	0.92	1.06	1.07	1.06	1.09	1.10
$9\times7\times64$	9	10	9	1.03	0.98	1.06	1.04	1.00	1.07	1.07	1.08	1.11
Total avg.			–			**1.01**			**1.02**			**1.08**

[a] Corresponding to the problem instances for which a feasible solution can be found by using *LINGO* within the time limit.

Table 6
Average running times of the SLR heuristic (s).

Size ($\|N\|\times\|T\|\times\|S\|$)	C_1	C_2	C_3
$6\times3\times4$	0.06	0.05	0.07
$6\times5\times16$	0.54	0.56	0.48
$6\times7\times64$	4.59	4.66	4.92
$7\times3\times4$	0.04	0.04	0.07
$7\times5\times16$	0.49	0.55	0.64
$7\times7\times64$	5.54	7.27	5.83
$8\times3\times4$	0.05	0.07	0.05
$8\times5\times16$	0.53	0.62	0.72
$8\times7\times64$	5.73	7.48	7.43
$9\times3\times4$	0.06	0.04	0.08
$9\times5\times16$	0.58	0.78	0.81
$9\times7\times64$	7.16	7.72	9.65
Avg.	**2.11**	**2.49**	**2.56**
Total avg.			**2.39**

LB^* and indicates the potential of the heuristic for moderate or large-scale problem instances.

(4) In Table 5, columns 5–7 list the average cost ratios of g_O^{LINGO} to LB^* under different problem configurations and the total average is 1.01. A comparison between the total average values of g_O^{LINGO}/LB^* and g_O^{SLR}/LB^* indicates that g_O^{SLR} is close to g_O^{LINGO}.

(5) The last three columns in both Tables 4 and 5 list the average cost ratios of UB^* to LB^* under different problem configurations. These ratios are close to 1, which means UB^* is close to LB^* and shows again the good performance of the LR algorithm for problem AP.

(6) From Table 4, a comparison between the average values of g_O^{opt}/LB^* and UB^*/LB^* shows that there is only a small deviation between them. This comparison reflects the good approximation of problem AP to problem OP through using the stepwise linearization.

(7) From Table 5, the total average value of g_O^{SLR}/LB^* is 1.02 while that of UB^*/LB^* is 1.08. The deviation between them is 6% while the corresponding deviation in Table 4 is 8%. This indicates that the SLR heuristic is promising and robust.

Table 6 reports the average running times of the SLR heuristic. The results show that the running time increases lineally as the problem size increases. For all the instances, the longest average running time is within 10 s. For an off-line production planning problem, planners are more concerned about the solution quality and they usually spend several hours on making manual production planning in practice. Therefore the running time of the heuristic is acceptable for the practical industrial size problem.

4.5. A discussion on scenario generation

In this section, we further discuss scenario generation to illustrate how the changes on demand uncertainty impact the solution. The change on demand uncertainty can be reflected by two aspects including the number of scenarios and the fluctuation range of demands. For the first aspect, by changing the number of scenarios of approach D mentioned in Section 4.1, we obtain two new scenario tree structures. One includes less number of scenarios than that of approach D and has an extreme structure where there is one branch out of each node and the associated $|S|$ is equal to 1. The other includes more number of scenarios than that of approach D. At the beginning of periods from $\lceil (|T|+3)/2\rceil$ to $|T|$, there are three branches out of each node. The branches at other nodes are the same as those in D. The associated $|S|$ is equal to $2^{\lfloor |T|/2\rfloor}\cdot 3^{\lceil |T|/2\rceil -1}$. For the second aspect, we obtain two new distributions with different fluctuation ranges through changing the fluctuation range of demands in approach D with the expectation of demands unchanged. One is a narrowed uniform distribution, a single-point distribution, R_2: {1100}. The other is an widened uniform distribution, R_3: $U(800, 1400)$. Except for the generation of the demand parameters, other parameters are generated by using the same approach as that in Section 4.1.

The computational results on the impact of different number of scenarios on the solutions of problem instances are reported in Table 7 while Table 8 reports the average costs of problem instances with different fluctuation range of demands. From the two tables, we have the following observations.

(1) The average cost increases as the number of scenarios increases or the fluctuation range of demand extends. These results show how the changes in demand uncertainty impact the production plan. If the uncertainty of the demand increases, the average cost increases. This indicates that more possible realizations or a greater variety of demands will result in additional costs and making the demand information more accurate will help reduce the production cost.

(2) The average running time increases lineally as the number of scenarios increases. This is because the increase in the number of scenarios increases the size of the problem.

Table 7
Average costs and running times for problem instances with different numbers of scenarios.

Setup cost	$\|N\|\times\|T\|$	Average cost		
		$\|S\|=1$	$\|S\|=2^{\|T\|-1}$	$\|S\|=2^{\lfloor \|T\|/2\rfloor}\cdot 3^{\lceil \|T\|/2\rceil -1}$
C_1	6×3	11,516,624	11,533,546	11,785,587
	6×5	19,280,522	19,299,428	19,496,347
	6×7	26,756,497	26,886,400	27,115,906
	7×3	13,469,126	13,535,330	13,493,721
	7×5	22,079,515	22,402,785	22,241,053
	7×7	30,802,335	31,346,142	31,340,278
	8×3	15,394,776	15,389,838	15,449,532
	8×5	25,584,637	25,341,851	25,520,400
	8×7	35,287,231	35,764,784	35,941,233
	9×3	17,209,188	17,348,891	17,447,308
	9×5	28,315,839	29,041,525	28,815,389
	9×7	39,732,955	39,962,663	40,049,904
Average cost with C_1		**23,785,770**	**23,987,765**	**24,058,055**
C_2	6×3	11,700,429	11,930,375	12,007,021
	6×5	19,293,127	19,684,271	19,917,771
	6×7	27,040,798	27,518,027	27,389,504
	7×3	13,541,065	13,796,298	13,799,368
	7×5	22,202,294	22,599,825	23,011,185
	7×7	30,960,712	31,798,443	31,668,106
	8×3	15,420,864	15,544,602	15,668,810
	8×5	25,279,810	25,824,984	26,112,068
	8×7	35,572,097	36,116,694	36,189,676
	9×3	17,375,688	17,758,560	17,789,692
	9×5	28,821,050	29,135,516	29,439,931
	9×7	40,343,813	41,244,229	41,243,051
Average cost with C_2		**23,962,646**	**24,412,652**	**24,519,682**
C_3	6×3	11,875,240	12,104,448	12,118,713
	6×5	19,636,163	20,019,758	20,144,460
	6×7	27,137,790	27,912,680	27,987,639
	7×3	13,763,802	13,985,233	14,078,447
	7×5	22,815,597	23,199,397	23,206,924
	7×7	31,808,611	32,454,720	32,468,473
	8×3	15,748,057	16,063,245	15,967,819
	8×5	25,906,318	26,251,728	26,618,507
	8×7	36,100,965	37,100,611	37,211,787
	9×3	17,880,791	17,964,968	18,293,124
	9×5	29,332,241	29,806,402	30,008,913
	9×7	40,708,039	41,688,541	41,661,545
Average cost with C_3		**24,392,801**	**24,879,311**	**24,980,529**
Total average cost		**24,047,072**	**24,426,576**	**24,519,422**
Total average running time (s)		**0.05**	**2.39**	**16.70**

Table 8
Average costs for problem instances with different fluctuation ranges of demands.

Setup cost	Size ($\|N\| \times \|T\| \times \|S\|$)	Average cost		
		R_2: {1100}	R_1: U(1000, 1200)	R_3: U(800, 1400)
C_1	$6 \times 3 \times 4$	11,728,966	11,533,546	11,932,265
	$6 \times 5 \times 16$	19,223,636	19,299,428	19,389,267
	$6 \times 7 \times 64$	26,778,677	26,886,400	26,910,482
	$7 \times 3 \times 4$	13,496,076	13,535,330	13,477,874
	$7 \times 5 \times 16$	22,343,393	22,402,785	22,548,281
	$7 \times 7 \times 64$	30,835,364	31,346,142	31,362,968
	$8 \times 3 \times 4$	15,160,618	15,389,838	15,373,072
	$8 \times 5 \times 16$	25,410,920	25,341,851	25,803,509
	$8 \times 7 \times 64$	35,358,020	35,764,784	35,631,978
	$9 \times 3 \times 4$	17,176,814	17,348,891	17,358,464
	$9 \times 5 \times 16$	28,411,941	29,041,525	28,429,225
	$9 \times 7 \times 64$	40,163,795	39,962,663	40,332,084
Average cost with C_1		**23,840,685**	**23,987,765**	**24,045,789**
C_2	$6 \times 3 \times 4$	11,801,952	11,930,375	11,853,393
	$6 \times 5 \times 16$	19,314,768	19,684,271	19,707,280
	$6 \times 7 \times 64$	27,554,317	27,518,027	27,022,897
	$7 \times 3 \times 4$	13,567,814	13,796,298	13,786,364
	$7 \times 5 \times 16$	22,405,397	22,599,825	22,893,642
	$7 \times 7 \times 64$	31,281,081	31,798,443	31,552,923
	$8 \times 3 \times 4$	15,444,516	15,544,602	15,505,303
	$8 \times 5 \times 16$	25,564,715	25,824,984	26,011,583
	$8 \times 7 \times 64$	35,768,879	36,116,694	36,346,815
	$9 \times 3 \times 4$	17,438,347	17,758,560	17,525,275
	$9 \times 5 \times 16$	28,953,118	29,135,516	29601,396
	$9 \times 7 \times 64$	40,324,760	41,244,229	41,533,753
Average cost with C_2		**24,118,305**	**24,412,652**	**24,445,052**
C_3	$6 \times 3 \times 4$	11,841,834	12,104,448	12,199,410
	$6 \times 5 \times 16$	19,627,264	20,019,758	20,034,213
	$6 \times 7 \times 64$	27,378,108	27,912,680	27,976,378
	$7 \times 3 \times 4$	13,714,464	13,985,233	13,952,439
	$7 \times 5 \times 16$	22,841,375	23,199,397	23,220,860
	$7 \times 7 \times 64$	31,703,389	32,454,720	32,541,297
	$8 \times 3 \times 4$	15,739,635	16,063,245	16,050,423
	$8 \times 5 \times 16$	26,116,009	26,251,728	26,371,002
	$8 \times 7 \times 64$	36,237,759	37,100,611	37,062,604
	$9 \times 3 \times 4$	17,893,476	17,964,968	17,994,875
	$9 \times 5 \times 16$	29,379,254	29,806,402	30,154,246
	$9 \times 7 \times 64$	41,142,599	41,688,541	41,634,555
Average cost with C_3		**24,467,931**	**24,879,311**	**24,932,692**
Total average cost		**24,142,307**	**24,426,576**	**24,474,511**

5. Conclusions

This paper investigates a stochastic production planning problem with nonlinear cost in the steel production process. The problem is characterized by its nonlinear objective and randomization of the demands. For the problem, we established the model by taking a scenario-based approach and developed a SLR heuristic. The computational experiment indicates that the proposed heuristic is promising for the problem. The discussion on scenario generation reflects that improving accuracy of demand information would have a positive influence on a rational decision.

Future attempts can be considered in the following aspects. First, the model can be extended to a more complex multi-stage inventory situation. Second, the proposed approach can be generalized to solve other stochastic production planning problems with nonlinear costs.

Acknowledgement

This research is partly supported by State Key Program of National Natural Science Foundation of China (71032004), National Natural Science Foundation for Distinguished Young Scholars of China (Grant no. 70728001), the Fundamental Research Funds for the Central Universities (N090104002) and the 111 Project under Grant B08015. The authors would like to thank the Production Manufacturing Center in Baoshan Iron and Steel Company for providing a lot of production information and data.

References

[1] Azaron A, Tang O, Tavakkoli-Moghaddam R. Dynamic lot sizing problem with continuous-time Markovian production cost. International Journal of Production Economics 2009;120:607–12.
[2] Bahl HC, Ritzman LP, Gupta JND. Determining lot sizes and resource requirements: a review. Operations Research 1987;35:329–45.
[3] Bakir MA, Byrne MD. Stochastic linear optimization of an MPMP production planning model. International Journal of Production Economics 1998;55:87–96.
[4] Barcia P, Jörnsten K. Improved Lagrangean decomposition: an application to the generalized assignment problem. European Journal of Operational Research 1990;46:84–92.
[5] Brandimarte P. Multi-item capacitated lot-sizing with demand uncertainty. International Journal of Production Research 2006;44:2997–3022.
[6] Carøe CC, Schultz R. Dual decomposition in stochastic integer programming. Operations Research Letters 1999;24:37–45.
[7] Chazal M, Jouini E, Tahraoui R. Production planning and inventories optimization: a backward approach in the convex storage cost case. Journal of Mathematical Economics 2008;44:997–1023.
[8] Chen Z, Li S, Tirupati D. A scenario based stochastic programming approach for technology and capacity planning. Computers and Operations Research 2002;29:781–806.
[9] Gutiérrez J, Puerto J, Sicilia J. The multiscenario lot size problem with concave costs. European Journal of Operational Research 2004;156:162–82.
[10] Haugen KK, Løkketangen A, Woodruff DL. Progressive hedging as a meta-heuristic applied to stochastic lot-sizing. European Journal of Operational Research 2001;132:116–22.
[11] Jans R, Degraeve Z. Meta-heuristics for dynamic lot sizing: a review and comparison of solution approaches. European Journal of Operational Research 2007;177:1855–75.
[12] Jörnsten K, Näsberg M. A new Lagrangian relaxation approach to the generalized assignment problem. European Journal of Operational Research 1986;27:313–23.
[13] Karimi B, Fatemi Ghomi SMT, Wilson JM. The capacitated lot sizing problem: a review of models and algorithms. Omega 2003;31:365–78.
[14] Minner S. A comparison of simple heuristics for multi-product dynamic demand lot-sizing with limited warehouse capacity. International Journal of Production Economics 2009;118:305–10.
[15] Mula J, Poler R, García-Sabater JP, Lario FC. Models for production planning under uncertainty: a review. International Journal of Production Economics 2006;103:271–85.
[16] Oh HC, Karimi IA. Planning production on a single processor with sequence-dependent setups part 1: determination of campaigns. Computers and Chemical Engineering 2001;25:1021–30.
[17] Rizk N, Martel A, Ramudhin A. A Lagrangean relaxation algorithm for multi-item lot-sizing problems with joint piecewise linear resource costs. International Journal of Production Economics 2006;102:344–57.
[18] Rockafellar RT, Wets RJB. Scenarios and policy aggregation in optimization under uncertainty. Mathematics of Operations Research 1991;16:119–47.
[19] Sox CR. Dynamic lot sizing with random demand and non-stationary costs. Operations Research Letters 1997;20:155–64.
[20] Takriti S, Birge JR. Lagrangian solution techniques and bounds for loosely coupled mixed-integer stochastic programs. Operations Research 2000;48:91–8.
[21] Tang L, Liu J, Rong A, Yang Z. A review of planning and scheduling systems and methods for integrated steel production. European Journal of Operational Research 2001;133:1–20.
[22] Tempelmeier H, Derstroff M. A Lagrangean-based heuristic for dynamic multilevel multiitem constrained lotsizing with setup times. Management Science 1996;42(5):738–56.
[23] Wagner HM, Whitin TM. Dynamic version of the economic lot size model. Management Science 1958;5:89–96.
[24] Xie J, Dong J. Heuristic genetic algorithms for general capacitated lot-sizing problems. Computers and Mathematics with Applications 2002;44:263–76.
[25] Yu S, Du T, Nan M, Han C, Ai M. A discussion of energy saving direction for rolling heating furnace. Industrial Furnace 1997;4:39–43.