Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2012, Article ID 582323, 21 pages
doi:10.1155/2012/582323

Research Article

Dynamic Programming and Heuristic for Stochastic Uncapacitated Lot-Sizing Problems with Incremental Quantity Discount

Yuli Zhang,[1] Shiji Song,[1] Cheng Wu,[1] and Wenjun Yin[2]

[1] *Department of Automation, TNList, Tsinghua University, Beijing 100084, China*
[2] *Industry Solutions, IBM Research-China, Beijing 100193, China*

Correspondence should be addressed to Shiji Song, shijis@mail.tsinghua.edu.cn

Received 30 January 2012; Revised 25 April 2012; Accepted 8 May 2012

Academic Editor: Jung-Fa Tsai

The stochastic uncapacitated lot-sizing problems with incremental quantity discount have been studied in this paper. First, a multistage stochastic mixed integer model is established by the scenario analysis approach and an equivalent reformulation is obtained through proper relaxation under the decreasing unit order price assumption. The proposed reformulation allows us to extend the production-path property to this framework, and furthermore we provide a more accurate characterization of the optimal solution. Then, a backward dynamic programming algorithm is developed to obtain the optimal solution and considering its exponential computation complexity in term of time stages, we design a new rolling horizon heuristic based on the proposed property. Comparisons with the commercial solver CPLEX and other heuristics indicate better performance of our proposed algorithms in both quality of solution and run time.

1. Introduction

The lot-sizing problems have been the subject of intensive research in the last few decades. The basic definition of single-item lot-sizing problems can be stated as follows: the order (production), inventory and backlog quantities in each period should be determined to meet the deterministic but dynamic demand over a finite time horizon. The objective is to minimize the total costs, which consist of the fixed setup cost, order cost and inventory cost. Different quantity discount policies, such as all-units quantity discount and incremental quantity discount, have been widely executed in practice and thus have also been introduced into the lot-sizing problems.

Although the deterministic planning and scheduling models have been intensively studied, in practice there are many different sources of uncertainties, such as customer demand, production lead-time, and product price, which make information that will be needed in subsequent decision stages unavailable to the decision maker. In such cases, the solution provided by a deterministic model may be of little value in terms of applicability of the model's recommendations, see Beraldi et al. [1]. Thus the stochastic version of lot-sizing problems has been studied recently and with the advent of stochastic programming, the classical deterministic lot-sizing models have been extended to scenario-based multistage stochastic mixed integer programming.

Wagner and Whitin [2] first introduced the single-item dynamic lot-sizing problems without backlogging. They proposed a dynamic programming approach based on the Wagner-Whitin (W-W) property, that is, no production is undertaken if inventory is available. Because W-W property holds for the nondiscount problem, many heuristics have been developed based on this property; however, in the all-units quantity discount case, the property does not necessarily hold due to its discontinuous piecewise linear cost structure by Hu and Munson [3]. For the modified all-units discount problem, Chan et al. [4] demonstrated that a zero-inventory-ordering policy based on the W-W property exists, whose cost is no more than 4/3 times the optimal cost. Federgruen and Lee [5] characterized structural properties of optimal solutions for both all-units and incremental quantity discount, and proposed dynamic programming algorithms with complexity $O(T^3)$ and $O(T^2)$, respectively, and with T being the number of periods in the planning horizon. For deterministic capacitated lot-sizing problem with general piecewise linear cost function, Shaw and Wagelmans [6] presented a dynamic programming procedure with complexity $O(T^2\overline{q}\overline{d})$, where T is the number of periods, $\overline{d}$ is the average demand and $\overline{q}$ is the average number of pieces required to represent the production cost function. Note that this pseudopolynomial time algorithm is based on the assumption that the demand is an integral value.

For the stochastic version problem, although Ahmed et al. [7] showed that the W-W property does not hold for the stochastic lot-sizing problems several modified W-W properties have been presented for different versions of the stochastic lot-sizing problems: Guan and Miller [8] studied the stochastic uncapacitated lot-sizing problems with uncertain parameters and introduced the production path property for the optimal solution. Further, the production path property was extended to the stochastic capacitated lot-sizing problems with backlogging in Guan and Miller [8] and the dynamic programming algorithms based on this property have been presented. For the stochastic lot-sizing problems with random lead times, Huang and Küçükyavuz [9] presented the Semi-Wagner-Whitin property under assumption that no order crosses in time.

Besides the algorithms based on the extended W-W properties, Lulli and Sen [10] and Ahmed et al. [7] have presented branch-and-price and branch-and-bound algorithms, respectively to solve the proposed multistage integer programming. Although such branch-and-bound-based (B&B-based) methods have broad application prospects for general integer programming, special structure properties of the stochastic lot-sizing problems have not been explored in order to design customized algorithms, and only computational results for small-size stochastic batch-sizing problems have been reported in Lulli and Sen [10]. Other heuristic methods, such as the fix and relax, have also been redesigned to solve particular stochastic lot-sizing problems, see [1, 11]. We also refer the reader to the recent literature review of [12–14].

To the best of our knowledge, little research has been reported on the stochastic lot-sizing problems with incremental quantity discount (SLP-IQD). However as it is reported in the survey by Munson and Rosenblatt [15], 83% of the buyers received quantity discounts for most of the items they purchased and 37% of orders involved either the offer or receipt of some incremental quantity discounts; thus, the study on quantity discount is of great importance in practice. In this paper, a multistage stochastic mixed-integer programming model is established and under decreasing unit order price assumption, an equivalent relaxed formulation is obtained. The reformulation provides the possibility of extending the production path property for optimal solution of SLP-IQD. The extended production path property is not only a direct extension to the case with incremental quantity discount, but also provides a more accurate characterization for the optimal solution. Then, a backward dynamic programming algorithm has been developed. Although it can obtain optimal solutions in polynomial time in terms of the number of nodes, it has an exponential computational complexity in terms of the number of stages. Thus, a new rolling horizon heuristic which makes use of the extended production path property and has flexible parameters settings is presented in order to balance the desired solution quality and run time. Numerical experiments have been implemented to explore the proper parameters settings and validate the effectiveness of the proposed algorithms by comparison with the CPLEX 11.1 solver and other heuristics.

The remainder of the paper is organized as follows. In Section 2, we first introduce the deterministic formulation and then formulate the general multistage stochastic mixed integer model for the stochastic uncapacitated lot-sizing problems with incremental quantity discount (SULP-IQD). An equivalent reformulation is proposed under the decreasing unit order price assumption. In Section 3, the extended production path property is proven and a backward dynamic programming algorithm and a rolling horizon heuristic with flexible parameters settings are developed. Computational results are reported and discussed in Section 4. Section 5 presents conclusions.

2. Mathematical Model

2.1. Deterministic Lot-Sizing Problems with Incremental Quantity Discount

First, we will establish an mathematical model for the deterministic uncapacitated lot-sizing problems with incremental quantity discount (DULS-IQD). Considering a planning horizon of T time periods (stages), at each period t, the nonnegative demand d_t, variable inventory holding cost h_t, the variable setup cost c_t, and piecewise order cost function $f_t(\cdot)$ at period $t \in \{1, \dots, T\}$ are given. Variable s_t denotes the inventory quantity at the end of period t, and variable x_t and Boolean variable y_t denote the order quantity and fixed charge variable at period t, respectively.

The incremental quantity discount cost structure is given as follows:

$$\text{the unit order cost} = \begin{cases} p_{t,1} & \text{for the first } Q_{t,1} \text{ units,} \\ p_{t,2} & \text{for the next } Q_{t,2} - Q_{t,1} \text{units,} \\ \vdots & \vdots \\ p_{t,K_t} & \text{for each unit in excess of } Q_{t,K_t-1} \text{units,} \end{cases} \tag{2.1}$$

where K_t denotes the number of price intervals. Suppose that $Q_{t,0} = 0$ and $Q_{t,K_t} = +\infty$. The *decreasing unit order price* assumption, that is, $p_{t,k} > p_{t,k+1}$, $k = 1, \dots, K_t - 1$ with

$0 = Q_{t,0} < Q_{t,1} < \cdots < Q_{t,k} < Q_{t,k+1} < \cdots < Q_{t,K_t} = +\infty$ always naturally holds in practice. The piecewise order cost function is given as

$$f_t(x_t) = \begin{cases} p_{t,1}x_t, & \text{if } Q_{t,0} \leq x_t < Q_{t,1}, \\ \sum_{j=1}^{k-1}\{(p_{t,j} - p_{t,j+1})Q_{t,j}\} + p_{t,k}x_t, & \text{if } Q_{t,k-1} \leq x_t < Q_{t,k},\ k = 2,\ldots,K_t. \end{cases} \tag{2.2}$$

Thus, the DULS-IQD can be formulated as

$$\min : \sum_{t=1}^{T}\{f_t(x_t) + h_t s_t + c_t y_t\} \tag{2.3}$$

subject to

$$s_{t-1} + x_t - d_t = s_t, \quad t = 1,\ldots,T, \tag{2.4}$$

$$x_t \leq M y_t, \quad t = 1,\ldots,T, \tag{2.5}$$

$$x_t \geq 0,\ s_t \geq 0,\ s_0 = 0 \quad t = 1,\ldots,T, \tag{2.6}$$

$$y_t \in \{0,1\}, \quad t = 1,\ldots,T. \tag{2.7}$$

The objective function (2.3) minimizes the sum of inventory cost, setup cost and piecewise order cost. Constraints (2.4) guarantee that the dynamic demands in every period are met. Constraints (2.5) are the capacity constraints, and here we assume that M is a sufficiently large upper bound on x_t. Constraints (2.6) ensure that there is no backlogging and order quantity variables are nonnegative. Without loss of generality, we suppose that there is no initial inventory. Setup variable y_t is defined as a binary variable in constraints (2.7).

2.2. Stochastic Lot-Sizing Problems with Incremental Quantity Discount

In this subsection, a stochastic model is established by scenario analysis approach. We assume that the problem parameters follow discrete time stochastic processes with a finite probability space and evolve as a multistage stochastic scenario tree. Let node n at stage t $(t = 1,\ldots,T)$ represent all the realization of the system leading up to and including stage t, and set N denotes the set of all nodes. Let set L denote the set of leaf nodes. Thus a scenario represents a path from the root node (indexed as $n = 0$) to a leaf node. The related notation is given as follows:

$a(n)$: parent of node n,

$t(n)$: time stage of node n, $t(n) \in \{1,\ldots,T\}$,

$D(n)$: the set of all descendants of node n and $n \in D(n)$,

$P(n,m)$: the set of nodes on a path from node n to node m, where $m \in D(n)$, and we assume that $P(n,m)$ includes n and m,

$C(n)$: children nodes of node n, that is, $C(n) = \{m \in N : a(m) = n\}$,

θ_n: the likelihood (probability) assigned to node n, $\theta_n > 0$.

Here using the above notation, the deterministic model in the above subsection can be extended to the stochastic environments by replacing the stage index $t \in \{1, \ldots, T\}$ with node index $n \in N$. The multistage stochastic problem formulation, denoted by SP, is given as follows:

$$\text{(SP)}\ \ \min : \sum_{n \in N} \{\theta_n \{f_n(x_n) + h_n s_n + c_n y_n\}\} \tag{2.8}$$

subject to

$$\begin{gathered} s_{a(n)} + x_n - d_n = s_n, \quad n \in N, \\ x_n \leq M y_n, \quad n \in N, \\ x_n \geq 0,\ s_n \geq 0,\ s_{a(0)} = 0, \quad n \in N, \\ y_n \in \{0, 1\}, \quad n \in N. \end{gathered} \tag{2.9}$$

An equivalent mixed integer programming formulation can be easily obtained by introducing auxiliary order quantity variables and corresponding Boolean variables. A group of variables are assigned for each order quantity interval as follows:

$$\begin{gathered} x_{n,k} \geq 0, \quad n \in N,\ k = 1, \ldots, K_n, \\ y_{n,k} = \begin{cases} 0, & x_{n,k} = 0, \\ 1, & Q_{n,k-1} < x_{n,k} \leq Q_{n,k}. \end{cases} \end{gathered} \tag{2.10}$$

For model brevity, we introduce the following notations: $\overline{p}_{n,k} \triangleq \theta_n p_{n,k}$, $\overline{h}_n \triangleq \theta_n h_n$, and $\overline{c}_{n,k} \triangleq \theta_n \{c_n + \sum_{j=1}^{k-1} \{(p_{n,j} - p_{n,j+1}) Q_{n,j}\}\}$. Note that the setup cost also increases correspondingly in index k, that is, $\overline{c}_{n,k} < \overline{c}_{n,k+1}$ for each node $n \in N$. An equivalent MIP formulation, ESP1, is given as follows:

$$\text{(ESP1)}\ \ \min : \sum_{n \in N} \left\{ \sum_{k=1}^{K_n} \left\{ \overline{p}_{n,k} x_{n,k} + \overline{c}_{n,k} y_{n,k} \right\} + \overline{h}_n s_n \right\} \tag{2.11}$$

subject to

$$s_{a(n)} + \sum_{k=1}^{K_n} x_{n,k} - d_n = s_n, \quad n \in N, \tag{2.12}$$

$$Q_{n,k-1} y_{n,k} \le x_{n,k} \le Q_{n,k} y_{n,k}, \quad n \in N, k = 1, \dots, K_n, \tag{2.13}$$

$$\sum_{k=1}^{K_n} y_{n,k} \le 1, \quad n \in N, \tag{2.14}$$

$$x_{n,k} \ge 0,\ s_n \ge 0,\ s_{a(0)} = 0, \quad n \in N, k = 1, \dots, K_n, \tag{2.15}$$

$$y_{n,k} \in \{0, 1\}, \quad n \in N, k = 1, \dots, K_n. \tag{2.16}$$

Proposition 2.1. *Formulation ESP1 is equivalent to the original formulation SP.*

Proof. Suppose that $\{x^*_{n,k}, y^*_{n,k}, s^*_n \mid n \in N, k = 1, \dots, K_n\}$ is an optimal solution of ESP1. For each $n \in N$, there exists at most one $k_n \in \{1, \dots, K_n\}$ such that $y^*_{n,k_n} = 1$ and $M \ge x^*_{n,k_n} > 0$ by constraints (2.13), (2.14), and the optimality. Note that if $\sum_{k'=1}^{K_n} x^*_{n,k'} = x^*_{n,k_n} = 0$, then $\sum_{k'=1}^{K_n} y^*_{n,k'} = y^*_{n,k_n} = 0$. Thus, by checking constraints in SP, we construct a corresponding feasible solution of SP as $\{\overline{x}^*_n = x^*_{n,k_n}, \overline{y}^*_n = y^*_{n,k_n}, \overline{s}^*_n = s^*_n \mid n \in N\}$. By the definition of $\overline{p}_{n,k}$, $\overline{c}_{n,k}$ and $\overline{h}_n$, the constructed feasible solution of SP has the same objective cost to $\{x^*_{n,k}, y^*_{n,k}, s^*_n \mid n \in N, k = 1, \dots, K_n\}$.

For each optimal solution of SP, $\{x^*_n, y^*_n, s^*_n \mid n \in N\}$, we define the corresponding solution for ESP1 as follows: for given $n \in N$, if $Q_{n,k-1} < x^*_n \le Q_{n,k}$, then $\{\overline{x}^*_{n,k} = x^*_n, \overline{y}^*_{n,k} = 1, \overline{s}^*_n = s^*_n, \overline{x}^*_{n,k'} = \overline{y}^*_{n,k'} = 0, k' \neq k\}$ and if $x^*_n = 0$, then $\{\overline{x}^*_{n,k} = 0, \overline{y}^*_{n,k} = 0, \overline{s}^*_n = s^*_n, k \in \{1, \dots, K_n\}\}$. By checking its feasibility and the objective cost in ESP1, we conclude that the constructed solution is also feasible in ESP1 and has same objective cost to $\{x^*_n, y^*_n, s^*_n \mid n \in N\}$. □

Further, by relaxing the constraints on Boolean variables and order quantity variables in (2.13)-(2.14), the following formulation, denoted by ESP2, and proposition can be obtained.

$$\text{(ESP2)}\ \ \min : \sum_{n \in N} \left\{ \sum_{k=1}^{K_n} \left\{ \overline{p}_{n,k} x_{n,k} + \overline{c}_{n,k} y_{n,k} \right\} + \overline{h}_n s_n \right\} \tag{2.17}$$

subject to

$$\begin{gathered} s_{a(n)} + \sum_{k=1}^{K_n} x_{n,k} - d_n = s_n, \quad n \in N, \\ x_{n,k} \le y_{n,k} M, \quad n \in N,\ k = 1, \dots, K_n, \\ x_{n,k} \ge 0,\ s_n \ge 0,\ s_{a(0)} = 0 \quad n \in N,\ k = 1, \dots, K_n, \\ y_{n,k} \in \{0, 1\}, \qquad n \in N,\ k = 1, \dots, K_n. \end{gathered} \tag{2.18}$$

Proposition 2.2. *Under the decreasing unit order price assumption, formulation ESP2 is equivalent to ESP1 and SP.*

Proof. Because the feasible set of the relaxed formulation ESP2 contains the feasible set of ESP1, the optimal solutions of ESP2 must be proven feasible for ESP1. Let $\{\tilde{x}^*_{n,k}, \tilde{y}^*_{n,k}, \tilde{s}^*_n \mid n \in N, k = 1, \dots, K_n\}$ be an optimal solution of ESP2. First note that if $\tilde{y}^*_{n,k} = 1$, we must have $\tilde{x}^*_{n,k} > 0$ otherwise it contradicts the optimality since $\bar{c}_{n,k} > 0$.

Second, it is asserted that the relaxed constraints $\sum_{k=1}^{K_n} y_{n,k} \le 1, n \in N$ always hold for every optimal solution of ESP2. Suppose that $\tilde{y}^*_{n,l} = \tilde{y}^*_{n,m} = 1, n \in N, 1 \le l \neq m \le K_n$; thus, $\tilde{x}^*_{n,l} > 0$ and $\tilde{x}^*_{n,m} > 0$. Without loss of generality, suppose that $m > l$ and $Q_{n,r-1} \le \tilde{x}^*_{n,l} + \tilde{x}^*_{n,m} < Q_{n,r}, 1 \le r \le K_n$, then

$$\begin{aligned}
&\bar{c}_{n,r} + \bar{p}_{n,r}\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) \\
&\quad = \theta_n \left\{ c_n + \sum_{j=1}^{r-1} \left\{ (p_{n,j} - p_{n,j+1}) Q_{n,j} \right\} + p_{n,r}\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) \right\} \\
&\quad = \theta_n \left\{ \sum_{j=m}^{r-1} \left\{ (p_{n,j} - p_{n,j+1}) Q_{n,j} \right\} + p_{n,r}\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) - p_{n,m}\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) \right\} \\
&\qquad + \bar{c}_{n,m} + \bar{p}_{n,m}\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) \\
&\quad \le \theta_n \left\{ \sum_{j=m}^{r-1} \left\{ (p_{n,j} - p_{n,j+1})\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) \right\} + p_{n,r}\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) - p_{n,m}\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) \right\} \\
&\qquad + \bar{c}_{n,m} + \bar{p}_{n,m}\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) \\
&\quad = \bar{c}_{n,m} + \bar{p}_{n,m}\left(\tilde{x}^*_{n,l} + \tilde{x}^*_{n,m}\right) \\
&\quad < \bar{c}_{n,m} + \bar{p}_{n,m}\tilde{x}^*_{n,m} + \bar{p}_{n,l}\tilde{x}^*_{n,l} \\
&\quad < \bar{c}_{n,m} + \bar{p}_{n,m}\tilde{x}^*_{n,m} + \bar{c}_{n,l} + \bar{p}_{n,l}\tilde{x}^*_{n,l},
\end{aligned} \tag{2.19}$$

where $\sum_{j=m}^{r-1} \{(p_{n,j} - p_{n,j+1}) = p_{n,m} - p_{n,r}$ in the third inequality.

From the above analysis, a better solution can always be obtained by setting $\hat{x}^*_{n,r} = \sum_{j=1}^{K_n} \tilde{x}^*_{n,j}, \hat{y}^*_{n,r} = 1$, where $Q_{n,r-1} \le \sum_{j=1}^{K_n} \tilde{x}^*_{n,j} \le Q_{n,r}$, and $\hat{x}^*_{n,k} = 0, \hat{y}^*_{n,k} = 0, k \neq r$, which contradicts the optimality.

Third, we prove that the relaxed constraints $Q_{n,k-1} y_{n,k} < x_{n,k} \le Q_{n,k} y_{n,k}, n \in N, k = 1, \dots, K_n$ also hold for the optimal solution. Assume that $\tilde{x}^*_{n,l} > 0$ and $\tilde{x}^*_{n,m} = 0$ for all $m \neq l$, but the inequality constraint $Q_{n,l-1} \leq \tilde{x}^*_{n,l} \leq Q_{n,l}$ does not hold. Without loss of generality, suppose that $Q_{n,r-1} < \tilde{x}^*_{n,l} \leq Q_{n,r} \leq Q_{n,l-1}$ where $r \leq l-1$, then

$$\begin{aligned}\overline{c}_{n,l} + \overline{p}_{n,l}\tilde{x}^*_{n,l} &= \theta_n \left\{ c_n + \sum_{j=1}^{l-1} \left\{ \left(p_{n,j} - p_{n,j+1}\right) Q_{n,j} \right\} + p_{n,l}\tilde{x}^*_{n,l} \right\} \\ &= \overline{c}_{n,r} + \overline{p}_{n,r}\tilde{x}^*_{n,l} + \theta_n \left\{ \sum_{j=r}^{l-1} \left\{ \left(p_{n,j} - p_{n,j+1}\right) Q_{n,j} \right\} + \left(p_{n,l} - p_{n,r}\right)\tilde{x}^*_{n,l} \right\} \\ &> \overline{c}_{n,r} + \overline{p}_{n,r}\tilde{x}^*_{n,l}.\end{aligned} \tag{2.20}$$

Therefore, we reach a contradiction because we obtain a better solution by setting $\hat{x}^*_{n,r} = \tilde{x}^*_{n,l}$, $\hat{y}^*_{n,r} = 1$, and $\hat{x}^*_{n,k} = 0$, $\hat{y}^*_{n,k} = 0$ for $k \neq r$. Since the optimal solution of ESP2 satisfies all the constraints of ESP1, and ESP2 is obtained from ESP1 by relaxing its constraints, this implies that ESP2 is equivalent to ESP1 and SP. □

3. Optimality Condition and Algorithms

In this section, we explore the property of the SULS-IQD and design-efficient algorithms. It is necessary to highlight the differences between the deterministic problems and the stochastic problems. First, Ahmed et al. [7] and Huang and Küçükyavuz [9] have shown that W-W property does not hold for the stochastic version problems. The reason is that one production or positive order, which is made to exactly cover the demand along certain path, will inevitably influence all its descendant nodes. The violation happens when this positive order could not cover some nodes and their inventory level is nonzero. Thus, we develop an extended production-path property for the stochastic problem and design a dynamic programming algorithm. Second, since the number of nodes grows exponentially as the number of time stages increases for stochastic version problems, the traditional dynamic programming methods fail to run efficiently. So, we further design an approximate heuristic based on the proposed property to obtain good enough solutions efficiently in this section.

3.1. Extended Production-Path Property

Formulation ESP2 presented in Section 2 is essential to handle the piecewise linear objective function and extend the production path property to the SULS-IQD. The following proposition is not only a direct extension of Guan and Miller [8] when incremental quantity discount is provided, but also a more accurate characterization of the optimal solution. It shows that there always exists an optimal solution such that if a positive order is made at node n, the order quantity exactly covers the demand along the path from node n to a certain descendant node m; moreover, no positive order will be made for these nodes along the path from node n to the parent node of m.

Proposition 3.1 (extended production-path property). *For any instance of SULS-IDQ, there exists an optimal solution (x^*, y^*, s^*), such that for each node $n \in N$, if $x^*_{n,k} > 0$, then there exist $m \in D(n)$ and $Q_{n,k-1} < d_{n,m} - s^*_{a(n)} \leq Q_{n,k}$ such that*

(1) $x^*_{n,i} = 0, y^*_{n,i} = 0$, *for all* $i = 1, \ldots, K_n, i \neq k$;

(2) $x^*_{n,k} = d_{n,m} - s^*_{a(n)}, y^*_{n,k} = 1$;

(3) $x^*_{l,i} = 0, y^*_{l,i} = 0$, *for all* $l \in P(n, a(m)) \setminus \{n\}, i = 1, \ldots, K_l$.

Note that under assumption that all lead time is equal to 1 and by similar arguments, the second optimal condition can be regarded as an extension of the Semi-Wagner-Whitin Property in Huang and Küçükyavuz [9]; however, the third optimal condition provides new restrictions which narrow the scope of the expected optimal solutions. In the next section, an improved backward dynamic programming algorithm and a rolling horizon heuristic based on the extended production-path property will be presented.

For any optimal solution of ESP2, (x, y, s), if $x_{n,k} > 0$, we introduce the following definition:

$$\begin{gathered} N_P(n) = \left\{ m \mid m \in D(n), \sum_{k=1}^{K_l} x_{l,k} = 0, \forall l \in P(n, a(m)) \setminus \{n\},\ \sum_{k=1}^{K_m} x_{m,k} > 0 \right\}, \\ N_{NP}(n) = \left\{ m \mid m \in D(n) \cap L, \sum_{k=1}^{K_l} x_{l,k} = 0, \forall l \in P(n, m) \setminus \{n\} \right\}, \end{gathered} \tag{3.1}$$

where set $N_P(n)$ contains the first node with positive order quantity after node n, and set $N_{NP}(n)$ contains these leaf nodes from which to node n no positive order has been placed except node n. Note that if a certain positive order quantity is properly transferred between node n and all the nodes in set $N_P(n)$, the feasibility can be kept and only the cost for nodes in the following set is changed:

$$\phi(n) = \{ m \mid m \in P(n, l), \forall l \in N_P(n) \cup N_{NP}(n) \}. \tag{3.2}$$

Proof. First, it is asserted that there exists at least an optimal solution by Weierstrass' theorem since the feasible set is compact (note that $\sum_{k=1}^{K_n} x_{n,k}$ can be bounded by $\max_{m \in D(n) \cup L} \{d_{n,m}\}$) and the objective function is continuous. Then for any given optimal solution of ESP2, (x^*, y^*, s^*), if $x^*_{n,k} > 0$, the first part holds from Proposition 2.2. Next, we show that an optimal solution holding the above property can always be constructed from any given optimal solution of ESP2 by adjusting the variable's value.

We scan the optimal solution (x^*, y^*, s^*) from stage 1 to stage T. Assume that there exists a node n with $x^*_{n,k} > 0$ not holding the property, but for nodes at stage $t < t(n)$ the conclusion holds, the following approach adjusts variables' value assigned to nodes in $D(n)$ to satisfy the property without changing variables' value before stage $t(n)$. For any feasible solution (x, y, s) of ESP2, we define the object cost function as

$$\mathcal{F}(x, y, s) = \sum_{n \in N} \left\{ \sum_{k=1}^{K_n} \left\{ \overline{p}_{n,k} x_{n,k} + \overline{c}_{n,k} y_{n,k} \right\} + \overline{h}_n s_n \right\}. \tag{3.3}$$

The objective cost for (x^*, y^*, s^*) is

$$
\begin{aligned}
&\mathcal{F}(x^*, y^*, s^*) \\
&\quad = \sum_{m \in N \setminus \phi(n)} \left\{ \sum_{j=1}^{K_m} \left\{ \overline{p}_{m,j} x^*_{m,j} + \overline{c}_{m,j} y^*_{m,j} \right\} + \overline{h}_m s^*_m \right\} + \overline{p}_{n,k} x^*_{n,k} + \overline{c}_{n,k} y^*_{n,k} \\
&\qquad + \overline{h}_n s^*_n + \sum_{m \in \phi(n) \{\{n\} \cup N_P(n)\}} \overline{h}_m s^*_m + \sum_{m \in N_P(n)} \left\{ \overline{p}_{m,j_m} x^*_{m,j_m} + \overline{c}_{m,j_m} y^*_{m,j_m} + \overline{h}_m s^*_m \right\},
\end{aligned}
\tag{3.4}
$$

where j_m is the unique positive order quantity at node $m, m \in N_P(n)$, and the node set N is divided into four subsets: $\{n\}$, $N_P(n)$, $\phi(n) \setminus \{\{n\} \cup N_P(n)\}$, and $N \setminus \phi(n)$. For any node $m \in \phi(n) \setminus \{\{n\} \cup N_P(n)\}$, no order is placed.

Since $x^*_{m,j_m} > 0$ for $m \in N_P(n)$ (by definition) and $s^*_m > 0$ for $m \in N_{NP}(n)$ (otherwise it contradicts the assumption), we choose small positive scalar ε satisfying

$$0 < \varepsilon \leq x^*_{n,k}, \tag{3.5}$$

$$0 < \varepsilon \leq s^*_m, \quad \text{for } m \in N_{NP}(n), \tag{3.6}$$

$$0 < \varepsilon \leq x^*_{m,j_m}, \quad \text{for } m \in N_P(n), \tag{3.7}$$

such that the following solutions are feasible for ESP2

$$
\begin{aligned}
\overline{x}^*_{n,k} = x^*_{n,k} - \varepsilon, \qquad \overline{s}^*_m = s^*_m - \varepsilon \quad \text{for } m \in \phi(n) \setminus N_P(n), \\
\overline{x}^*_{m,j_m} = x^*_{m,j_m} + \varepsilon \quad \text{for } n \in N_P(n),
\end{aligned}
\tag{3.8}
$$

$$
\begin{aligned}
\widehat{x}^*_{n,k} = x^*_{n,k} + \varepsilon, \qquad \widehat{s}^*_m = s^*_m + \varepsilon \quad \text{for } m \in \phi(n) \setminus N_P(n), \\
\widehat{x}^*_{m,j_m} = x^*_{m,j_m} - \varepsilon \quad \text{for } n \in N_P(n)
\end{aligned}
\tag{3.9}
$$

and keep the value of other variables unchanged. For given ε satisfying constraints (3.5)–(3.7), the objective costs of $(\overline{x}^*, \overline{y}^*, \overline{s}^*)$ and $(\widehat{x}^*, \widehat{y}^*, \widehat{s}^*)$ are

$$
\begin{aligned}
&\mathcal{F}(\overline{x}^*, \overline{y}^*, \overline{s}^*) \\
&\quad = \sum_{m \in N \setminus \phi(n)} \left\{ \sum_{j=1}^{K_m} \left\{ \overline{p}_{m,j} x^*_{m,j} + \overline{c}_{m,j} y^*_{m,j} \right\} + \overline{h}_m s^*_m \right\} + \overline{p}_{n,k} \left(x^*_{n,k} - \varepsilon \right) + \overline{c}_{n,k} y^*_{n,k} + \overline{h}_n (s^*_n - \varepsilon) \\
&\qquad + \sum_{m \in \phi(n) \setminus \{\{n\} \cup N_P(n)\}} \overline{h}_m (s^*_m - \varepsilon) + \sum_{m \in N_P(n)} \left\{ \overline{p}_{m,j_m} \left(x^*_{m,j_m} + \varepsilon \right) + \overline{c}_{m,j_m} y^*_{m,j_m} + \overline{h}_m s^*_m \right\} \\
&\quad = \mathcal{F}(x^*, y^*, s^*) - \overline{p}_{n,k} \varepsilon - \overline{h}_n \varepsilon - \sum_{m \in \phi(n) \setminus \{\{n\} \cup N_P(n)\}} \overline{h}_m \varepsilon + \sum_{m \in N_P(n)} \overline{p}_{m,j_m} \varepsilon \\
&\quad = \mathcal{F}(x^*, y^*, s^*) - \overline{p}_{n,k} \varepsilon - \sum_{m \in \phi(n) \setminus N_P(n)} \overline{h}_m \varepsilon + \sum_{m \in N_P(n)} \overline{p}_{m,j_m} \varepsilon,
\end{aligned}
$$

$$\begin{aligned}
&\mathcal{F}(\widehat{x}^*, \widehat{y}^*, \widehat{s}^*)\\
&\quad= \sum_{m\in N\backslash\phi(n)} \left\{ \sum_{j=1}^{K_m} \left\{ \overline{p}_{m,j} x^*_{m,j} + \overline{c}_{m,j} y^*_{m,j} \right\} + \overline{h}_m s^*_m \right\} + \overline{p}_{n,k}\left(x^*_{n,k}+\varepsilon\right) + \overline{c}_{n,k} y^*_{n,k} + \overline{h}_n (s^*_n+\varepsilon)\\
&\qquad + \sum_{m\in\phi(n)\backslash\{\{n\}\cup N_P(n)\}} \overline{h}_m(s^*_m+\varepsilon) + \sum_{m\in N_P(n)} \left\{\overline{p}_{m,j_m}\left(x^*_{m,j_m}-\varepsilon\right) + \overline{c}_{m,j_m} y^*_{m,j_m} + \overline{h}_m s^*_m\right\}\\
&\quad= \mathcal{F}(x^*,y^*,s^*) + \overline{p}_{n,k}\varepsilon + \overline{h}_n\varepsilon + \sum_{m\in\phi(n)\backslash\{\{n\}\cup N_P(n)\}}\overline{h}_m\varepsilon - \sum_{m\in N_P(n)} \overline{p}_{m,j_m}\varepsilon.\\
&\quad= \mathcal{F}(x^*,y^*,s^*) + \overline{p}_{n,k}\varepsilon + \sum_{m\in\phi(n)\backslash N_P(n)}\overline{h}_m\varepsilon - \sum_{m\in N_P(n)} \overline{p}_{m,j_m}\varepsilon.
\end{aligned} \tag{3.10}$$

Note that the first equality comes from the definition of $(\overline{x}^*, \overline{y}^*, \overline{s}^*)$, $(\widehat{x}^*, \widehat{y}^*, \widehat{s}^*)$, and $\mathcal{F}$ function. The second one is obtained by comparing with the value of $\mathcal{F}(x^*, y^*, s^*)$ and the last one is obtained by rearranging the terms in the former, where $\overline{h}_n\varepsilon + \sum_{m\in\phi(n)\backslash\{\{n\}\cup N_P(n)\}}\overline{h}_m\varepsilon = \sum_{m\in\phi(n)\backslash N_P(n)}\overline{h}_m\varepsilon$.

Since (x^*, y^*, s^*) is an optimal solution of ESP2, thus we must have $\overline{p}_{n,k} + \sum_{m\in\phi(n)\backslash N_P(n)}\overline{h}_m - \sum_{m\in N_P(n)}\overline{p}_{m,j_m} = 0$ and $(\overline{x}^*, \overline{y}^*, \overline{s}^*)$, $(\widehat{x}^*, \widehat{y}^*, \widehat{s}^*)$ are also optimal solution for ESP2. Now we increase ε so that at least one of the following cases occur.

Case 1. If equality in (3.5) holds, then we have eliminated the undesired positive order at node n and $(\overline{x}^*, \overline{y}^*, \overline{s}^*)$ will be used to construct an eventual solution by a similar method.

Case 2. If equality in (3.6) for a certain m holds, then there exists $m \in D(n)$ such that $\overline{x}^*_{n,k} = d_{n,m} - s^*_{a(n)}$ and $\overline{x}^*_{l,k} = 0, \overline{y}^*_{l,k} = 0$ for all $l \in P(n, a(m)) \setminus \{n\}, k = 1, \dots, K_l$. Next, $(\overline{x}^*, \overline{y}^*, \overline{s}^*)$ will be used to construct an eventual solution by a similar method.

Case 3. If equality in (3.7) for a certain m holds, then we apply the above analysis to $(\widehat{x}^*, \widehat{y}^*, \widehat{s}^*)$. Since there are finite nodes in $D(m)$, eventually case 1 or case 2 will occur (in the worst case $N_P(n) = \emptyset$ after finite steps).

Thus, the optimal solution holding the proposed property can always be constructed after finite steps. □

3.2. Dynamic Programming Algorithm

To recursively calculate the optimal solution, the following functions are introduced as in Guan and Miller [8]:

$H(n, s)$: value function at node n when its initial inventory is $s_{a(n)} = s$, that is, $H(n, s) = \min\{\sum_{m\in D(n)}\{\sum_{k=1}^{K_m}\{\overline{p}_{m,k}x_{m,k} + \overline{c}_{m,k}y_{m,k}\} + \overline{h}_m s_m\}\}$ subject to constraints (2.18) and $s_{a(n)} = s$:

$H_P(n, s)$: production value function at n when its initial inventory is $s_{a(n)} = s$ and $\sum_{k=1}^{K_n} x_{n,k} > 0$:

$H_{NP}(n, s)$: nonproduction value function at n when its initial inventory is $s_{a(n)} = s$ and $\sum_{k=1}^{K_n} x_{n,k} = 0$.

```
For each stage $t = 1$ to $T$
    For each node $n$ at stage $t$
        For each possible initial inventory $s_m = d_{0,m} - d_{0,a(n)} \geq 0$, $m \in N$
        (if $t = 1$, set $s_m = 0$)
            step 1: Calculate $H_P(n, s_m)$ by (3.11)
            step 2: Calculate $H_{NP}(n, s_m)$ by (3.12) if $d_{0,m} - d_{0,n} >= 0$,
                otherwise $H_{NP}(n, s_m) = +\infty$
            step 3: Calculate $H(n)$ by (3.13)
        End For Iteration ($s_m$)
    End For Iteration ($n$)
End For Iteration ($t$)
```

Algorithm 1: Dynamic programming algorithm.

From Proposition 3.1, if a positive order quantity is made at node n when the initial inventory is s, then there exists a node $j \in D(n)$, such that $x_n = d_{n,j} - s$. Therefore, the following equations hold:

$$H_P(n,s)$$

$$= \min_{j \in D(n):\ Q_{n,k-1} < d_{n,j} - s \leq Q_{n,k}} \left\{ \overline{c}_{n,k} + \overline{p}_{n,k}\left(d_{n,j} - s\right) + h_n\left(d_{n,j} - d_n\right) + \sum_{m \in C(n)} H\left(m, d_{n,j} - d_n\right) \right\}, \tag{3.11}$$

$$H_{NP}(n,s) = h_n(s - d_n) + \sum_{m \in C(n)} H(m, s - d_n), \tag{3.12}$$

$$H(n,s) = \min\{H_P(n,s), H_{NP}(n,s)\}. \tag{3.13}$$

To obtain the exact optimal solution of SP, it is not necessary to completely characterize the value function $H(0,s)$. We only need to calculate its values at possible positive discontinuous points. That is, with zero initial inventory assumption for node $n \in N \setminus \{0\}$, we only need to evaluate $H(n,s)$ for $s = d_{0,m} - d_{0,a(n)} \geq 0, m \in N$ since $d_{n,m} - d_n = d_{0,m} - d_{0,n}$ in (3.11). Thus, we give the following dynamic programming for SULS-IQD:

Without loss of generality, we assume $\mid C(n) \mid = \mathcal{C}$ for all $n \in N$ in the following analysis. Dynamic programming demonstrates a straight method to obtain the optimal solution. Although Guan and Miller [8] showed that general SULS without incremental quantity discount can be solved in $O(\mid V(0)|^2 \max\{\log V(0), \mathcal{C}\})$, the above algorithm runs in exponential time in terms of T since $V(0) = (\mathcal{C}^T - 1)/(\mathcal{C} - 1)$. The exponential computational time encourages us to develop a more effective algorithm for the problems with large numbers of stages (see Algorithm 1).

3.3. Rolling Horizon Heuristic

In dynamic lot-sizing and scheduling problems with a large planning horizon, rolling horizon heuristics have been developed to decompose the original large-scale problem into a set of smaller subproblems. See, for example, [11, 16]. In contrast with the existing heuristics, the proposed RHH based on the extended production-path property stems from the proposed dynamic programming algorithm. In the heuristic, nonproduction strategy is developed to take advantage of the accurate characterization of the optimal solution, and then look-forward and look-backward heuristic strategies are developed to reduce the computation complexity of the complete enumeration in evaluation of the value function. The computation complexity is also analyzed to demonstrate its advantage for the problems with a large planning horizon.

(1) *Nonproduction Strategy*

At each iteration for s_m in DP algorithm for given n, if $m \in D(n) \setminus \{n\}$, step 1 can be skipped and set $H(n) = H_{NP}(n, s_m)$ since it violates part 3 of Proposition 3.1. In such case, even if $H(n) = H_P(n, s_m) < H_{NP}(n, s_m)$, and there exist optimal solutions with $\sum_{k=1}^{K_n} x_{n,k} > 0$, we will not lose the optimality because it is guaranteed that there exists an optimal solution satisfying the extended production path property in Proposition 3.1. The nonproduction strategy can be stated as

$$\text{if } m \in D(n) \setminus \{n\}, \text{then skip step1 and set } H(n) = H_{NP}(n, s_m) \text{ at step3.} \tag{3.14}$$

(2) *Look-Forward Strategy*

At step 1 for given node n at stage t and initial inventory $s_m = d_{0,m} - d_{0,a(n)}$, (3.11) requires calculating all the positive orders that cover demand from node n to a certain node $j \in D(n)$. These complete enumeration calculations are very time-consuming, thus we define the following look-forward strategy with parameter FT (forward time stage):

$$\begin{aligned} &H_P(n, s) \\ &\quad = \min_{j \in D_{\mathrm{FT}}(n): Q_{n,k-1} \le d_{n,j} - s \le Q_{n,k}} \left\{ \overline{c}_{n,k} + \overline{p}_{n,k}\left(d_{n,j} - s\right) + h_n\left(d_{n,j} - d_n\right) + \sum_{m \in C(n)} H\left(m, d_{0,j} - d_{0,n}\right) \right\}, \end{aligned} \tag{3.15}$$

where $D_{\mathrm{FT}}(n) = \{j \mid j \in D(n) \text{ and } t(j) - t(n) \le \mathrm{FT}\}$ denotes the set of node n's descendants within FT generations. The problem is how to select proper FT. It is obvious that increasing FT will not only improve the quality of solution, but will also increase computation time. Thus, the quality of solution and run time can be balanced by properly selecting FT. The performance of the proposed RHH with different FT settings will be tested by numerical experiments in the next section.

(3) *Look-Backward Strategy*

At step 2 for given node n at stage t and initial inventory $s_m = d_{0,m} - d_{0,a(n)}$, if stage $t(m)$ is smaller than t, it is likely $s_m - d_n = d_{0,m} - d_{0,n} < 0$ and in such cases step 2 can be skipped. For the same reason, if $t - t(m)$ is larger than a certain value, we can expect $s_m = d_{0,m} - d_{0,a(n)} < 0$

and this iteration for s_m can be skipped too. Look-backward strategy with parameter BF (backward time stage) can be stated as follows:

$$\begin{aligned}&\text{if } t-t(m)>BF, \text{ then skip this iteration for } s_m;\\ &\quad\text{else if } t-t(m)=BF, \text{ skip step2 and set } H(n)=H_P(n,s_m) \text{ at step 3,}\end{aligned} \tag{3.16}$$

where BF depends on the distribution of demand. We will set proper BF based on distribution of demand by numerical experiments in the next section.

In the above strategies, the major modification comes from the iteration for s_m and calculation of $H_P(n)$ at step 1 for given node n. In order to give a brief characterization of the above heuristic rules and analyze its computational complexity, we give the following lemma.

Lemma 3.2. *For each node $n \in N \setminus \{0\}$, the rolling horizon heuristic with FT and BT only needs to evaluate the following values:*

$$\{H(n, d_{0,m} - d_{0,a(n)}) : d_{0,m} - d_{0,a(n)} \geq 0, m \in RH_{FT,BT}(n)\}, \tag{3.17}$$

where $RH_{FT,BT}(n) = \cup_{m\in P(0,a(n)),t(n)-t(m)\leq FT+BT} D_{FT}(m)$.

Proof. This lemma is proven by induction from nodes at stage $t = 1$ to leaf nodes at stage T. Since positive order must be made at node 0 due to the zero initial inventory assumption, thus for node n at stage $t(n) = 2$ by (3.15) possible initial inventory $s_{a(n)}$ at node n could only be a certain value in set $\{d_{0,m}-d_{0,a(n)} : d_{0,m}-d_{0,a(n)} \geq 0, m \in \text{RH}_{\text{FT,BT}}(n)\} = \{d_{0,m}-d_{0,0} : d_{0,m}-d_{0,0} \geq 0, m \in D_{\text{FT}}(0)\}$. Suppose the lemma holds for node $a(n)$ at stage t, where $2 \leq t \leq T$, that is, the set of initial inventory for node $a(n)$ is given by $\{d_{0,m} - d_{0,a(a(n))} : d_{0,m} - d_{0,a(a(n))} \geq 0, m \in \text{RH}_{\text{FT,BT}}(a(n))\}$.

Now consider the possible initial inventory at node n.

Case 1. If a positive order has been made at node $a(n)$, by (3.15) in the look-forward strategy, the set of initial inventory at nodes n is given by $\{d_{0,m} - d_{0,a(n)} : d_{0,m} - d_{0,a(n)} \geq 0, m \in D_{FT}(a(n))\}$.

Case 2. if no order has been made at node $a(n)$, the set of possible initial inventory at node n is given by $\{d_{0,m} - d_{0,a(n)} : d_{0,m} - d_{0,a(n)} \geq 0, m \in \text{RH}_{\text{FT,BT}}(a(n))\}$. However, note that by look-backward strategy (3.16) we skip the calculation of $H_{\text{NP}}(a(n), d_{0,m} - d_{0,a(n)})$ for those m such that $t(a(n)) - t(m) = \text{BF}$; thus, in this case we only need to consider initial inventory value in set $\{d_{0,m} - d_{0,a(n)} \geq 0, m \in \cup\{D_{\text{FT}}(m) : m \in P(0, a(a(n))), t(a(n)) - t(m) \leq \text{FT} + \text{BT} - 1\}\} = \{d_{0,m} - d_{0,a(n)} \geq 0, m \in \cup\{D_{\text{FT}}(m) : m \in P(0, a(a(n))), t(n) - t(m) \leq \text{FT} + \text{BT}\}\}$ at node n.

Combine the above two cases, the conclusion holds for node n. □

Figure 1 gives an intuitive example of a balanced scenario tree with $C = 2$ at each node. For the rolling horizon heuristic with FT = 2 and BT = 1, given $\text{RHH}_{2,1}(a(n))$ for node $a(n)$, consider the initial inventory set for node n. If a positive order has been made at node $a(n)$, $\{d_{0,j} - d_{0,a(n)} : d_{0,j} - d_{0,a(n)} \geq 0, j \in D_{\text{FT}}(a(n))$ and $t(j) = t(n) + \text{FT} - 1\}$ will be introduced by (3.15). By the look-backward strategy (3.16), we do not need to calculate $H(n, s_m)$, where $t(m) = t(n) - \text{BT} - 1$. Other values will be inherited if no order has been made at node $a(n)$.

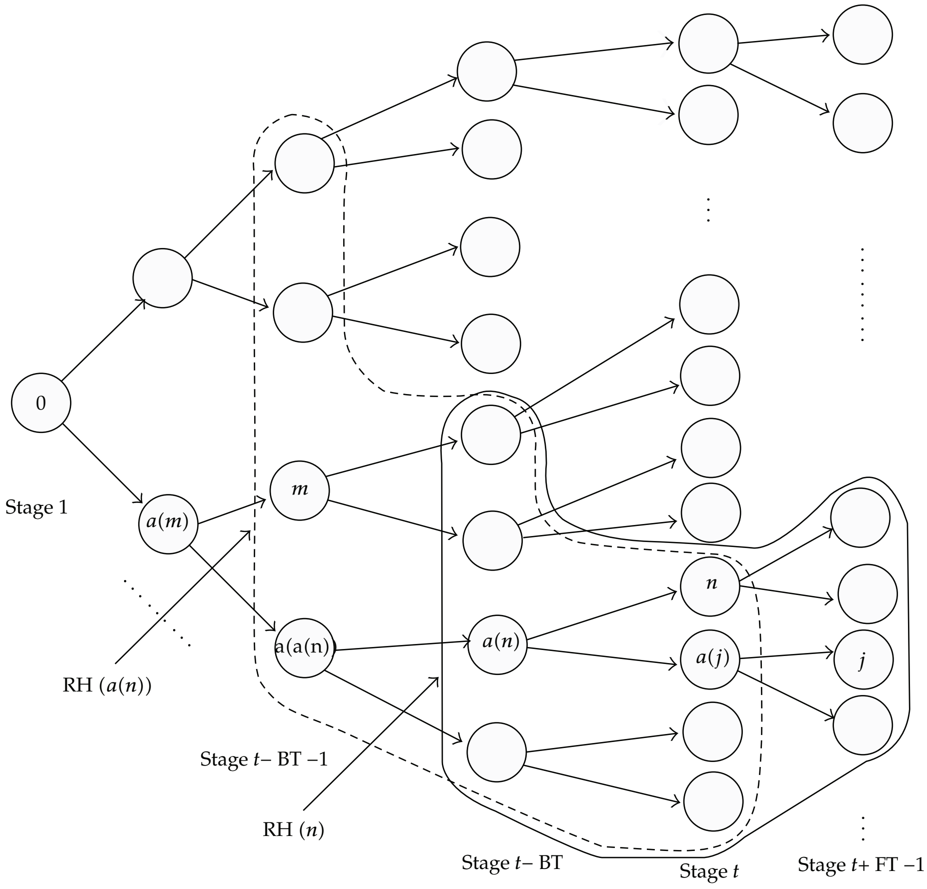


Figure 1: Example of RHH with FT = 2 and BT = 1.

Note that $\mid \mathrm{RH}_{\mathrm{FT,BT}}(n) \mid$ is no more than $(\mathrm{FT}+\mathrm{BT})\mathcal{C}^{\mathrm{FT}}$ where we assume that $\mid C(n) \mid = \mathcal{C}$ for all $n \in N$.

By summarizing the above analysis, the rolling horizon heuristic is given as in Algorithm 2.

Next the computation complexity of RHH is analyzed. For given node n at stage t, there exist at most $\mid \mathrm{RH}_{\mathrm{FT,BT}}(n) \mid$ possible initial inventories s_m. For each given s_m, it takes $O(K_n \mid C(n) \mid \mid D_{\mathrm{FT}}(n) \mid)$ time to complete calculation of $H_{\mathrm{P}}(n, s_m)$ at step 1. Step 2 and step 3 will be completed in $O(C(n))$ time. Thus the total run time can be estimated as (assume that $K_n = K$ for all $n \in N$)

$$\begin{aligned}\sum_{n\in N}\{K\mathcal{C}|\mathrm{RH}_{\mathrm{FT,BT}}(n)||D_{\mathrm{FT}}(n)|\} &\le K\mathcal{C}(\mathrm{FT}+\mathrm{BT})\mathcal{C}^{\mathrm{FT}} \sum_{n\in N}\{|D_{\mathrm{FT}}(n)|\} \\ &= K\mathcal{C}(\mathrm{FT}+\mathrm{BT})\mathcal{C}^{\mathrm{FT}} \frac{\mathcal{C}^{\mathrm{FT}+1}-1}{\mathcal{C}-1}|D(0)|\end{aligned}$$

```
For each stage $t = 1$ to $T$
    For each node $n$ at stage $t$
        For each initial inventory $s_m \in \{d_{0,m} - d_{0,a(n)} \geq 0 : m \in \mathrm{RH}_{\mathrm{FT,BT}}(n)\}$
        Set $H_P(n, s_m) = H_{NP}(n, s_m) = +\infty$; (if $t = 1$, set $s_m = 0$)
            step 1: If $m \in D(n) \setminus \{n\}$, go to step 3; otherwise calculate $H_P(n, s_m)$
                by (3.15)
            step 2: If $t - t(m) = \mathrm{BF}$ or $d_{0,n} - d_{0,m} < 0$, goto step 3; otherwise
                calculate $H_{NP}(n, s_m)$ by (3.12)
            step 3: Calculate $H(n)$ by (3.13)
        End For Iteration ($s_m$)
    End For Iteration ($n$)
End For Iteration ($t$)
```

Algorithm 2: Rolling horizon heuristic (FT, BT).

$$\begin{aligned} &= K(\mathrm{FT} + \mathrm{BT})\frac{\mathcal{C}^{\mathrm{FT}+1}}{\mathcal{C} - 1}\left\{\mathcal{C}^{\mathrm{FT}+1} - 1\right\}|D(0)| \\ &\leq K(\mathrm{FT} + \mathrm{BT})\mathcal{C}^{2(\mathrm{FT}+1)}|D(0)|. \end{aligned} \tag{3.18}$$

Proposition 3.3. *For any instance of SULS-IDQ, the rolling horizon heuristic with parameter FT and BT runs in $O(K\ (FT + BT)\ \mathcal{C}^{2(FT+1)} \mid D(0) \mid)$ time.*

The above analysis can be applied to the dynamic programming algorithm, thus the total run time for DP is given by

$$\begin{aligned} \sum_{n \in N}\{K\mathcal{C}|D(0)||D(n)|\} &= K\mathcal{C}|D(0)| \sum_{n \in N}\{|D(n)|\} \\ &= K\mathcal{C}|D(0)| \sum_{t=1}^{T}\mathcal{C}^{t-1}\left\{1 + \mathcal{C}^1 + \cdots + \mathcal{C}^{T-t}\right\} \\ &= K\ |D(0)|\ \frac{\mathcal{C}}{\mathcal{C} - 1}\left\{T\mathcal{C}^T - \frac{\mathcal{C}^T - 1}{\mathcal{C} - 1}\right\}. \end{aligned} \tag{3.19}$$

By comparison of computational complexity in (3.18) and (3.19), we observe that RHH works more efficiently for large T. Effective parameters settings of RHH will be further explored by numerical experiments in the next section.

4. Computational Experiment

In this section, the computational results on both DP and RHH are reported. In the computational analysis, we first concentrate on identifying proper settings of parameters FT and BT for RHH by comparison of its relative error gap and run time with DP's. Then, DP and RHH are compared with the CPLEX solver and other heuristics for the lot-sizing problems with fixed charge.

Table 1: Problem instances.

Instance	C	T	N	Int.	Cont.
$G1$	2	10	1023	3069	7161
$G2$	2	11	2047	6141	8188
$G3$	2	12	4095	12285	16380
$G4$	3	7	1093	3279	4372
$G5$	3	8	3280	9840	13120
$G6$	3	9	9841	29523	39564
$G7$	4	6	1365	4095	5460
$G8$	4	7	5461	16383	21844

Table 2: Problem parameters.

Parameter	Distribution
$\overline{h}_{n,k}$	Uniform $(1,2)$
$\overline{p}_{n,1}$	Uniform $(1,2)$
$\overline{c}_{n,1}$	Uniform $(3,5)$
$\overline{Q}_{n,k}$	Uniform $(5,15)$
$d_{n,k}$	Truncated normal $(8,\sigma^2)$

4.1. Problem Instance

In order to evaluate performance of the proposed DP and RHH, and explore the proper parameters settings of FT and BT for RHH, 8 groups of problem instances are generated by varying the number of stages from 5 to 12 and the number of branches from 2 to 4. The number of incremental quantity discount intervals is fixed to 3. Table 1 gives the number of branches C, the number of stages T, the number of nodes N, and the number of integer and continuous variables in formulation ESP2 for each group of problem instances. Other parameters are generated randomly and we assume that they are a sequence of i.i.d. random variables which are subject to truncated normal distribution or uniform distribution. We report in Table 2 part of the problem parameters and the variance of $d_{n,k}$ will be further defined by $\sigma = 8 * cv$ (coefficient of variation) in the following subsection. Due to decreasing unit order price assumption, the unit order price and setup cost at node n is generated by $\overline{p}_{n,k} = (1-\alpha)\overline{p}_{n,k-1}$ and $\overline{c}_{n,k} = \overline{c}_{n,k-1} + \alpha\overline{p}_{n,k-1}\overline{Q}_{n,k-1}$, where α is the discount factor and we set $\alpha = 0.05$. Our experiments are conducted on a workstation clocked at 2.33 GHz and equipped with 11.9 GB of core memory.

4.2. Numerical Results

In order to evaluate the performance of the proposed RHH method with different parameters, we define two different implementations of RHH and the problem parameter cv varies from 0.05 to 0.25 for each group of problems. RHH$_1$ with parameters FT $= \lceil T/3 \rceil$ and BT $= 1$ has been designed to obtain good enough solution in a short time, while RHH$_2$ with parameters FT $= \lceil T/2 \rceil$ and BT $= 2$ has been designed to obtain better solutions, where $\lceil x \rceil$ denotes the smallest integer larger than x. Table 3 gives the optimal objective function values and CPU

Table 3: Performance of RHH with different (FT, BT).

Instance	cv	RHH_1		RHH_2		DP	
		RE (%)	CPU (sec)	RE (%)	CPU (sec)	Value	CPU (sec)
$G1-1$	0.05	0.289	0.016	0.0	0.094	12968.28	0.100
$G1-2$	0.15	0.168	0.015	0.0	0.110	12991.93	0.985
$G1-3$	0.25	0.135	0.015	0.0	0.110	13082.23	0.969
$G2-1$	0.05	0.039	0.079	0.0	0.219	25694.23	4.672
$G2-2$	0.15	0.015	0.079	0.0	0.219	26111.22	4.672
$G2-3$	0.25	0.005	0.078	0.0	0.218	26447.11	4.703
$G3-1$	0.05	0.009	0.172	0.0	1.109	51470.76	22.047
$G3-2$	0.15	0.008	0.172	0.0	1.109	51722.50	22.109
$G3-3$	0.25	0.006	0.156	0.0	1.110	52772.90	22.172
$G4-1$	0.05	2.431	0.047	0.509	0.156	8322.42	0.656
$G4-2$	0.15	1.529	0.031	0.195	0.125	8998.02	0.641
$G4-3$	0.25	1.391	0.047	0.158	0.125	9126.63	0.640
$G5-1$	0.05	2.877	0.140	0.635	0.516	24898.40	7.344
$G5-2$	0.15	1.869	0.125	0.409	0.516	26456.16	7.359
$G5-3$	0.25	1.212	0.141	0.190	0.500	27716.36	7.375
$G6-1$	0.05	2.950	0.438	0.214	5.937	74120.22	81.484
$G6-2$	0.15	1.879	0.453	0.064	5.969	79096.31	81.563
$G6-3$	0.25	1.434	0.453	0.054	6.047	82641.24	81.844
$G7-1$	0.05	7.894	0.031	1.270	0.093	7814.61	0.828
$G7-2$	0.15	5.590	0.016	0.783	0.093	8551.74	0.813
$G7-3$	0.25	4.350	0.015	0.435	0.094	9024.62	0.812
$G8-1$	0.05	1.697	0.437	0.247	2.219	30614.77	16.969
$G8-2$	0.15	1.001	0.453	0.075	2.234	33982.89	16.938
$G8-3$	0.25	0.597	0.437	0.053	2.234	36099.06	17.032

time in seconds for DP, and the percentage relative error (RE(%)) and CPU time for RHH_1 and RHH_2, where

$$\text{RE} = \frac{\text{Value of RHH} - \text{Value of DP}}{\text{Value of DP}} \times 100\%. \tag{4.1}$$

For RHH_1, the computational results show that RHH_1 is able to solve all the instances in the shortest time with no more than 3% relative error except, for instance, $G7$. The solution quality and run time are further improved when the number of stages T becomes larger and the worst results of $G7$ can be interpreted as the look-forward time horizon being too small to obtain near optimal solution. For RHH_2, the relative error has been further decreased compared with RHH_1. Although the run time has been increased, RHH_2 still has advantage in run time compared with DP especially for large problem instances. We also note that as the coefficient of variation increases, the solutions quality and run time have been constantly improved for all problem instances.

Then in the second experiment, we concentrate on the comparison of solution quality and computation time with standard MIP solver CPLEX (version 11.1) and another heuristic dynamic slope scaling procedure (DSSP). DSSP proposed by Kim and Pardalos [17] is a

Table 4: Comparison with DSSP and CPLEX.

Instance	RHH		DSSP		CPLEX		DP	
	RE (%)	CPU (sec)	RE (%)	CPU (sec)	RE (%)	CPU (sec)	Value	CPU (sec)
$G1-1$	0.0	0.094	2.225	1.062	11.881	5.110	11452.16	0.985
$G1-2$	0.0	0.109	2.299	0.907	10.441	5.063	11653.62	0.984
$G1-3$	0.0	0.094	2.581	1.062	9.555	5.015	11685.286	0.984
$G2-1$	0.0	0.516	1.986	2.172	0.619	25.750	22602.25	4.719
$G2-2$	0.0	0.516	2.583	2.438	3.800	24.093	23408.59	4.719
$G2-3$	0.0	0.500	3.040	2.094	6.103	24.110	23259.73	4.719
$G3-1$	0.0	1.140	1.947	4.578	0.645	111.891	45820.49	22.187
$G3-2$	0.0	1.125	2.845	4.250	1.162	114.657	46665.20	22.172
$G3-3$	0.0	1.125	3.104	4.406	0.858	113.641	47492.82	22.218
$G4-1$	0.704	0.140	0.721	1.015	3.744	3.391	7285.71	0.656
$G4-2$	0.110	0.140	0.893	0.828	15.302	3.375	7916.02	0.656
$G4-3$	0.035	0.140	1.089	0.735	13.440	3.297	8279.95	0.641
$G5-1$	0.867	0.515	0.484	3.313	3.334	37.516	21619.82	7.453
$G5-2$	0.174	0.500	0.632	2.796	1.199	38.265	23767.47	7.422
$G5-3$	0.198	0.516	1.163	3.110	0.814	37.703	24543.10	7.438
$G6-1$	0.202	6.063	0.401	8.781	0.800	414.046	65133.43	82.640
$G6-2$	0.068	6.109	0.747	11.453	1.879	412.860	70243.82	82.375
$G6-3$	0.022	6.110	1.034	11.157	0.843	419.937	74203.32	82.563
$G7-1$	0.201	0.360	0.716	1.078	4.300	4.250	6681.52	0.828
$G7-2$	0.0	0.359	0.889	0.859	1.742	4.172	7480.93	0.812
$G7-3$	0.0	0.344	1.368	1.188	1.188	4.2347	7980.60	0.828
$G8-1$	0.367	2.204	0.429	4.578	3.918	86.125	26885.65	17.265
$G8-2$	0.082	2.172	0.579	4.812	0.741	87.125	30213.60	17.234
$G8-3$	0.0	2.171	1.060	5.594	1.111	89.468	32503.12	17.157

newly developed effective heuristic algorithm that provides good-quality solution to the concave piecewise linear optimization problem and among the heuristics it works well in practice [18] The initial solutions and updating schemes are implemented in accordance with the recommendations in [18] and the stopping criterion is given as follows: if $\|x^k - x^{k-1}\| \le \varepsilon$ or the iteration reaches MaxIteration, then it stops, where MaxIteration = 20. To compare the solution quality for given time limit, we set the time limit of CPLEX solver to 5 time of the total run time of DP. We report in Table 4 the percentage relative error and CPU time of RHH (its parameters are set according to RHH$_2$), DSSP and CPLEX and the optimal objective function value and CPU time of DP. Table 4 shows that the proposed RHH outperforms the DSSP in both the quality of solution and run time for almost all the test instances, and the CPLEX solver fails to find optimal solution within the given time.

In summary, the proposed DP can solve the SULP-IQD efficiently compared with the standard CPLEX solver and by properly setting the parameters, we obtain effective RHH which outperforms the DSSP heuristic for the tested instances. The computational results also show that RHH performs better for problem instances with a larger number of stages and high coefficient of variation.

5. Conclusion

In this paper, we study the stochastic uncapacitated lot-sizing problems with incremental quantity discount where the uncertain parameters are supposed to evolve as discrete time stochastic processes. First, we establish the original stochastic formulation by scenario analysis approach. Another two formulations are built by introducing auxiliary variables and relaxing certain constraints. Then, it is proven that under the decreasing unit order price assumption, the relaxed formulation is equivalent to the original one. Based on this reformulation, the extended production-path property is presented for the SULP-IQD and it enhances the ability to further refine the desired optimal solution by providing a more accurate characterization.

To obtain the exact optimal solution, a dynamic programming algorithm is developed. Although the dynamic programming algorithm has the polynomial-time computational complexity in terms of the number of nodes, it runs exponentially in terms of the number of stages. Thus, a new rolling horizon heuristic is further designed which contains three types of strategies to reduce the computational time. The nonproduction strategy is designed based on the accurate characterization of the optimal solution, and the look-forward and look-backward strategy is developed to overcome the complete enumeration calculations in the production and nonproduction value function. Numerical experiments are carried out to identify proper parameters settings of the proposed RHH and to evaluate the performance of the proposed algorithms by comparison with the CPLEX solver and DSSP heuristic. The computational results of a large group of problem instances with different parameters setting suggest that DP outperforms the CPLEX solver in run time required for obtaining optimal solution and the proposed RHH demonstrates satisfactory run time and solution quality compared with DSSP heuristic; moreover, as the computational complexity analysis suggests, the performance of RHH is better for problems with a greater number of stages.

Since the main difficulties for the stochastic lot-sizing problems stem from the setup cost and uncertain parameters, it will be an area of future research to analyze the properties of the problem and present effective algorithms for the stochastic lot-sizing problems with complex setup requirements, such as setup carryovers by Buschkühl et al. [14] and sequence-dependent setup costs by Beraldi et al. [1].

Acknowledgments

The paper is supported by the National Nature Science Foundation of China (NSFC no. 60874071 and 60834004) and Research Fund for the Doctoral Program of Higher Education of China (RFDP no. 20090002110035).

References

[1] P. Beraldi, G. Ghiani, A. Grieco, and E. Guerriero, "Fix and relax heuristic for a stochastic lot-sizing problem," *Computational Optimization and Applications*, vol. 33, no. 2-3, pp. 303–318, 2006.

[2] H. M. Wagner and T. M. Whitin, "Dynamic version of the economic lot size model," *Management Science*, vol. 5, pp. 89–96, 1958.

[3] J. Hu and C. L. Munson, "Dynamic demand lot-sizing rules for incremental quantity discounts," *Journal of the Operational Research Society*, vol. 53, no. 8, pp. 855–863, 2002.

[4] L. M. A. Chan, A. Muriel, Z. J. Shen, and D. S. Levi, "On the effectiveness of zero-inventory-ordering policies for the economic lot-sizing model with a class of piecewise linear cost structures," *Operations Research*, vol. 50, no. 6, pp. 1058–1067, 2002.

[5] A. Federgruen and C. Y. Lee, “The dynamic lot size model with quantity discount,” *Naval Research Logistics*, vol. 37, no. 5, pp. 707–713, 1990.

[6] D. X. Shaw and A. P. M. Wagelmans, “An algorithm for single-item capacitated economic lot sizing with piecewise linear production costs and general holding costs,” *Management Science*, vol. 44, no. 6, pp. 831–838, 1998.

[7] S. Ahmed, A. J. King, and G. Parija, “A multi-stage stochastic integer programming approach for capacity expansion under uncertainty,” *Journal of Global Optimization*, vol. 26, no. 1, pp. 3–24, 2003.

[8] Y. P. Guan and A. J. Miller, “Polynomial-time algorithms for stochastic uncapacitated lot-sizing problems,” *Operations Research*, vol. 56, no. 5, pp. 1172–1183, 2008.

[9] K. Huang and S. Küçükyavuz, “On stochastic lot-sizing problems with random lead times,” *Operations Research Letters*, vol. 36, no. 3, pp. 303–308, 2008.

[10] G. Lulli and S. Sen, “A branch-and-price algorithm for multistage stochastic integer programming with application to stochastic batch-sizing problems,” *Management Science*, vol. 50, no. 6, pp. 786–796, 2004.

[11] P. Beraldi, G. Ghiani, E. Guerriero, and A. Grieco, “Scenario-based planning for lot-sizing and scheduling with uncertain processing times,” *International Journal of Production Economics*, vol. 101, no. 1, pp. 140–149, 2006.

[12] D. Quadt and H. Kuhn, “Capacitated lot-sizing with extensions: a review,” *A Quarterly Journal of Operations Research*, vol. 6, no. 1, pp. 61–83, 2008.

[13] J. Raf and D. Zeger, “Modeling industrial lot sizing problems: a review,” *International Journal of Production Research*, vol. 46, no. 6, pp. 1619–1643, 2008.

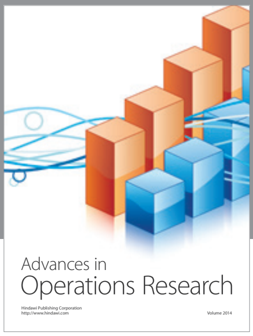
[14] L. Buschkühl, F. Sahling, S. Helber, and H. Tempelmeier, “Dynamic capacitated lot-sizing problems: a classification and review of solution approaches,” *OR Spectrum*, vol. 32, no. 2, pp. 231–261, 2010.

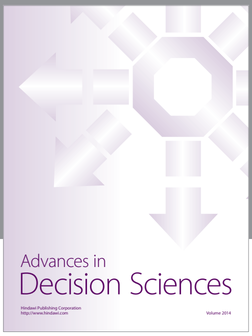
[15] C. L. Munson and M. J. Rosenblatt, “Theories and realities of quantity discounts: an exploratory study,” *Production and Operations Management*, vol. 7, no. 4, pp. 352–369, 1998.

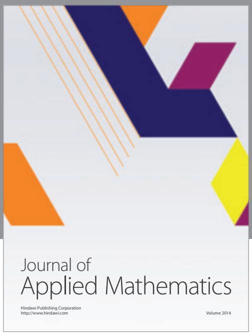
[16] P. Beraldi, G. Ghiani, A. Grieco, and E. Guerriero, “Rolling-horizon and fix-and-relax heuristics for the parallel machine lot-sizing and scheduling problem with sequence-dependent set-up costs,” *Computers and Operations Research*, vol. 35, no. 11, pp. 3644–3656, 2008.

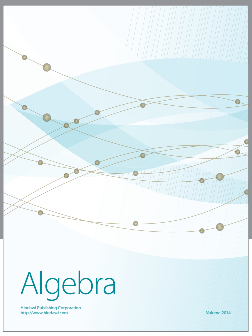
[17] D. Kim and P. M. Pardalos, “A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure,” *Operations Research Letters*, vol. 24, no. 4, pp. 195–203, 1999.

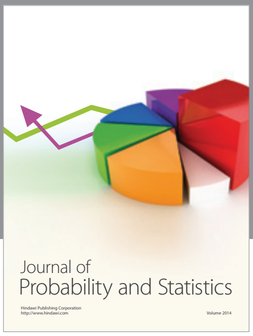
[18] S. Lawphongpanich, “Dynamic slope scaling procedure and lagrangian relaxation with subproblem approximation,” *Journal of Global Optimization*, vol. 35, no. 1, pp. 121–130, 2006.


Advances in
Operations Research

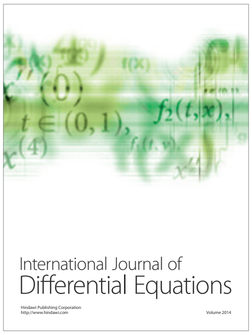

Advances in
Decision Sciences

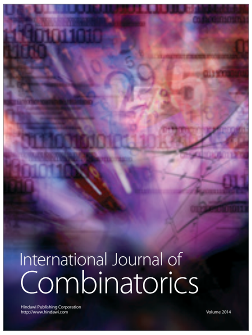

Journal of
Applied Mathematics

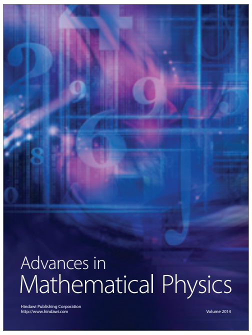

Algebra

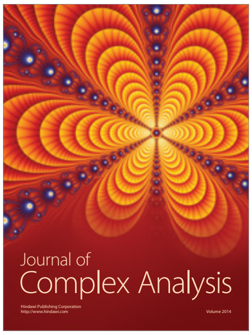

Journal of
Probability and Statistics

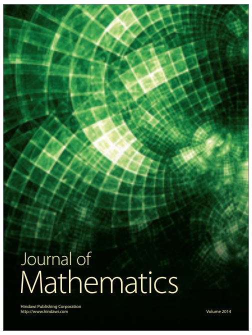
The Scientific
World Journal

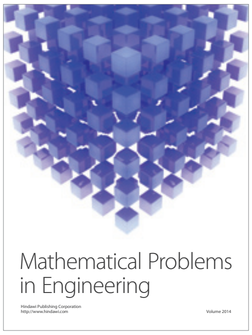

International Journal of
Differential Equations

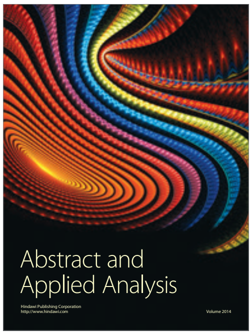

International Journal of
Combinatorics

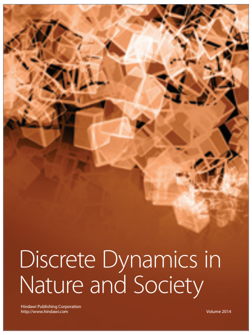

Advances in
Mathematical Physics

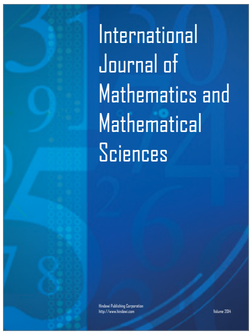

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

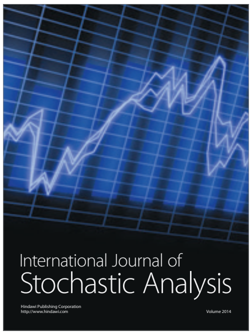

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization