

Research Article

Strategic Inventory Positioning in BOM with Multiple Parents Using ASR Lead Time

Jingjing Jiang and Suk-Chul Rim

Department of Industrial Engineering, Ajou University, San 5, Woncheon-dong, Yeongtong-gu, Suwon 443-749, Republic of Korea

Correspondence should be addressed to Jingjing Jiang; haha-jjj@163.com

Received 30 October 2015; Revised 4 January 2016; Accepted 7 February 2016

Academic Editor: Jei-Zheng Wu

Copyright © 2016 J. Jiang and S.-C. Rim. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to meet the lead time that the customers require, work-in-process inventory (WIPI) is necessary at almost every station in most make-to-order manufacturing. Depending on the station network configuration and lead time at each station, some of the WIPI do not contribute to reducing the manufacturing lead time of the final product at all. Therefore, it is important to identify the optimal set of stations to hold WIPI such that the total inventory holding cost is minimized, while the required due date for the final product is met. The authors have presented a model to determine the optimal position and quantity of WIPI for a given simple bill of material (S-BOM), in which any part in the BOM has only one immediate parent node. In this paper, we extend the previous study to the general BOM (G-BOM) in which parts in the BOM can have more than one immediate parent and present a new solution procedure using genetic algorithm.

1. Introduction

Since 1964, extensive investigations have been conducted on the materials requirement planning (MRP), which ensures materials to be available to meet manufacturing and market demands. Utilizing master production schedule (MPS) and bill of material (BOM) as input data, MRP determines when and how many items to produce or purchase to meet the MPS. MRP has been widely used in most manufacturing industry for generating a detailed production schedule and purchase order for two decades since introduced.

However, due to the rapid development of technologies, wider choice of customers, and lower forecast accuracy, the performance of MRP became unsatisfactory. In 2011, Ptak and Smith [1] developed a new type of MRP, named as demand-driven MRP (DDMRP), which replaces the previous convention of safety stock with strategically replenished positions. As an innovative multiechelon pull methodology, DDMRP can plan inventory of materials, enable a company to build plans more closely to actual market demands, and promote better and quicker decisions and actions in the planning and execution. Traditional MRP focuses on answering for how

much inventory to hold, and when to release order, while DDMRP focuses on answering for where to position the work-in-process inventory (WIPI).

The make-to-order (MTO) strategy makes the end product only when the customer places the order. The lead time for MTO is relatively longer compared to the make-to-stock (MTS) for the customer to get the product. In order to shorten the overall response time to the customer in MTO manufacturing, WIPI needs to be held in the station network. However, holding WIPI in some stations does not contribute to shortening the manufacturing lead time of the final product at all, only to increase the total inventory holding cost. So it is important to identify the optimal set of stations to place WIPI in the station network. We named this problem as the strategic inventory positioning (SIP) problem in our previous work (Rim et al. [2]). In the previous work, we address the SIP problem for the simple BOM (S-BOM) in which any part in the BOM has only one immediate parent part. The current study, however, is an extension or generalization of the previous work in that we address the SIP problem for the general BOM (G-BOM) in which some parts in the BOM have more than one immediate parent.

Only the SIP problem with BOM having a small number of parts can obtain the optimal solution by enumeration within a reasonable computation time. For S-BOM with n items, SIP problem is to choose a set of zero-one decisions out of 2^n alternatives, of which one must be optimal. However, for G-BOM, if one part has m parent parts, the number of alternatives needs to be multiplied by $m!$. As more items have multiple parents in the BOM, it is impractical to enumerate all solutions due to the excessively long computation time.

Therefore the objective of this paper is to develop a new mathematical model and a new genetic algorithm to solve the SIP problem for G-BOM, in which some parts in the BOM have more than one immediate parent. Our goal is to find a solution method that can balance the solution quality with the computational feasibility. We will present genetic operators that will limit the search to a specific set of feasible solutions instead of exploring all possible solutions.

This paper is organized as follows: a review of the related literature is presented in Section 2. ASRLT in S-BOM and G-BOM is introduced in Section 3. Section 4 is dedicated to the problem description and model formulation of the new SIP problem in DDMRP. The process of building the genetic algorithm to solve the problem and the computational results using the proposed GA are given in Section 5. Conclusion and future works are given in Section 6.

2. Literature Review

The inventory positioning problem has been studied by many researchers for past several years. In some previous studies, authors solved the problem of determining the optimal inventory position to increase the service fill rate and minimize the total inventory holding cost against a facility or full supply chains. In order to arrive the best service level, Whybark and Yang [3] proposed a controlled simulation experiment to decide where to place inventory. However, we determine the optimal inventory position depending on the replenishment model presented in DDMRP. Our paper has same targets including increasing service fill rate and minimizing total inventory holding cost with those in the previous literatures, but the method of calculating the lead time of the end product that the customer requires and the total inventory holding cost for all stations is different with those in previous research.

Many researchers apply Simpson's model to solve the inventory positioning problems. Simpson Jr. [4] used the "all or nothing" policy to decide whether to place the inventory or not in some points in serial line system. He indicated that the service time is independent variable, and the total inventory holding cost is the objective function. The safety stock for time t period is considered as the average inventory when he calculated the total inventory cost. He defined the safety stock as multiplying standard deviation of the demand for time t period by the safety factor.

Graves and Willems [5–7] extended Simpson Jr. [4] model to assembly, distribution, and spanning tree network structures. They employed the periodic review base-stock-system model assuming no capacity constraints. In their paper, they considered the service time as a decision variable

and minimizing the total inventory holding cost as the objective. They calculated the total inventory holding cost just as what Simpson Jr. [4] did but proposed DP algorithm.

Lesnaia [8] applied the Graves and Willems [6] model to the supply chain in a manufacturing firm. The difference is that Lesnaia considered the service time as a stochastic model instead of a deterministic model assumed by Graves and Willems. Lesnaia presented a general-network algorithm to manage the safety stock placement problem and determine the optimal service time for the path.

Magnanti et al. [9] solved the inventory positioning problem at production/assembly stages of components in an acyclic supply chain network structure which is not a spanning tree network structure. The service delivery time and inbound replenishment lead time are considered as decision variables and minimizing the total inventory cost is considered as the objective in all stages of the network. They built an efficient MIP formulation and employed a successive piecewise linear approximation approach.

Kaminsky and Kaya [10] developed an effective heuristic algorithm using linear programming modes to handle problems in supply chain including inventory positioning, short and reliable due-date quotation, and order sequencing. They defined two expected times as decision variables. One is that the safety stock of components stored at facility j that are received from upstream facility k for the production of type i will last; another one is that the inventory of finished goods at facility j required for the production of product type i will last. The objective function consists of inventory costs, average lead time costs, and tardiness cost in a facility.

Inderfurth [11] and Inderfurth and Minner [12] applied the Simpson model to divergent and convergent systems in the network structure. The paper used a periodic review base-stock control policy and considered minimizing the inventory cost as the objective function and the service time as a decision variable.

Some researchers adopted the postponement concept in determining the inventory position in the supply chain such as Ioannou et al. [13]. This idea was initially introduced by Alderson [14] and then has been studied by Bucklin [15], Lee et al. [16], Davis [17], Feitzinger and Lee [18], and Ernst and Kamrad [19].

Rim et al. [2] stated the total inventory cost as the objective function and the strategic inventory position as decision variable that is different from the service time presented in previous studies. The paper used the new and important concept of ASRLT introduced by Ptak and Smith [1]. A mathematical model is built and GA method is proposed to determine the inventory position in BOM to minimize the total inventory cost while meeting the lead time required by customers. In this paper, we continue using the basic theory brought by our last paper (Rim et al. [2]). However, the BOM structure considered in last paper is simplified, so we generalize the BOM structure in this paper so that the calculation of ASRLT becomes more complicated. The new mathematical model and GA method are proposed in this paper so that the SIP problem in DDMRP is studied completely in aspect of BOM structure.

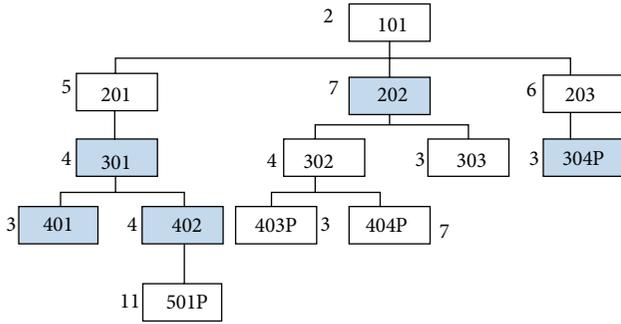


FIGURE 1: An example of simple bill of materials (S-BOM).

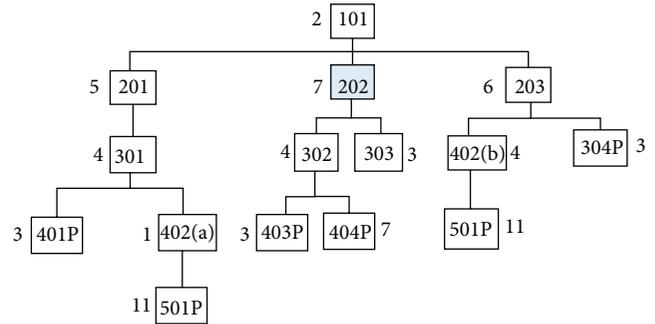


FIGURE 3: Sequence 1.

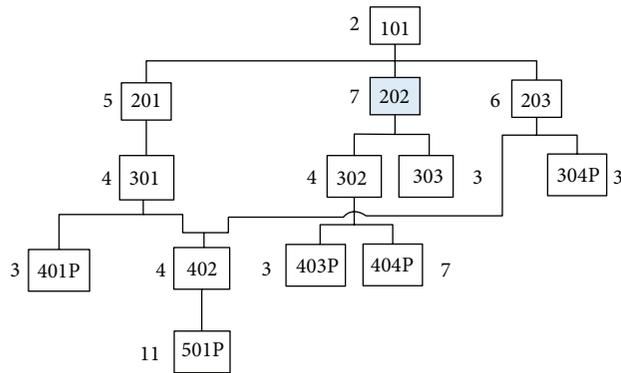


FIGURE 2: An example of general bill of materials (G-BOM).

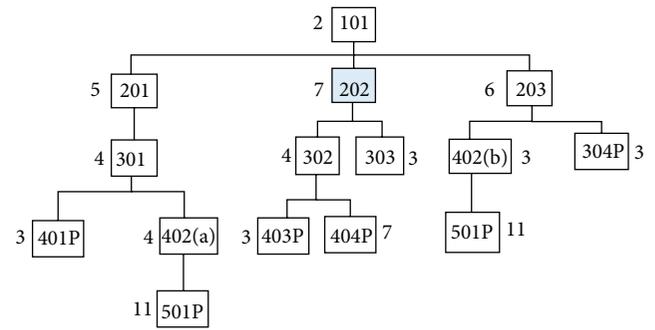


FIGURE 4: Sequence 2.

3. ASRLT in S-BOM and G-BOM

In this chapter, we introduce Actively Synchronized Replenishment (ASR) lead time. ASR lead time is defined as the longest unprotected or unbuffered sequence in the BOM for a particular parent (Ptak and Smith [1]). Figure 1 from Rim et al. [2] shows that the manufacturing lead time of part 101, the end product of the BOM, is 2 days, given that three components of level 2 are all available. The longest path to calculate the cumulative lead time is 26 days from 101 through 201, 301, 402, to 501P.

Suppose the five shaded nodes 202, 301, 304P, 401, and 402 of the BOM have stocks. Then the ASRLT of the end product is $2 + 6 = 8$ days. If part 203 additionally holds stock, the ASRLT of product 101 becomes $2 + 5 = 7$ days. The more the nodes we allocate the stock at, the shorter (or equal) the ASRLT will be, obviously with higher total inventory cost.

In the general BOM (G-BOM), some of parts have multiple immediate parents so that the processing of the part will be divided into several sections to meet the requirement of different parent parts. Since the ASRLT will change depending on which immediate parent part to supply first, the sequence is critical.

For example, part 402 in Figure 2 has two immediate parent parts 301 and 203. The processing of part 402 is divided into 2 sections: 402(a) for 301 and 402(b) for 203. Suppose that 2 units of 402(a) and 6 units of 402(b) are required to produce one unit of 101. Then the total processing times of 402(a) and 402(b) are 1 day ($= 4 * 2 / (2 + 6)$) and 3 days

($= 4 * 6 / (2 + 6)$), respectively. Depending on which to produce first, two different BOMs appear, as shown in Figures 3 and 4.

In Figure 3, 402(a) is processed first for 301 and the manufacturing lead time (MLT) of 402(a) is 1. However, 402(b) has to wait processing for 203 until 402(a) is finished, so the MLT of 402(b) is $1 + 3 = 4$ days. The ASRLT of 301 and 203 is $4 + 1 + 11 = 16$ and $6 + 4 + 11 = 21$, respectively. The common parent part of 301 and 203 is 101. (In this example, the common part happens to be the end product.) In order to just compare the ASRLT of the common parent part under these two sequences, suppose 202 has inventory. Then the ASRLT of 101 is $2 + \max(16, 21) = 23$ days. On the other hand, in Figure 4, 402(b) is processed first for 203, so the MLT of 402(b) and 402(a) is 3 and 4, respectively. So ASRLT of 101 is 26 days. By comparing the ASRLT of 101 using the two processing sequences, we find that processing 402(a) first shortens the MLT of 101.

4. The Strategic Inventory Positioning Problem

In the earlier study (Rim et al. [2]) we presented a mathematical model to determine the locations among the stations in the S-BOM which hold WIP inventory in order to meet the customer lead time, while minimizing the total inventory holding cost. We named this problem as the strategic inventory positioning (SIP) problem. In this section, we consider the SIP problem for the G-BOM, in which some nodes of the BOM may have two immediate parent nodes. To distinguish the two SIP problems, the earlier SIP problem, in which all

nodes have only one immediate parent node, is named as SIP-S (SIP in S-BOM) problem. The problem we address in this paper, in which some nodes may have multiple immediate parent nodes, is named as SIP-G (SIP in G-BOM) problem. For convenience, we will use the same notations that we used in our earlier paper and add a few new notations to describe the SIP-G problem as follows:

$\text{part}_{i,j}$: j th part counted from the left in the i th level of the BOM;

$p_{i,j} = \{(x, y)\}$: set of the two immediate parent parts of $\text{part}_{i,j}$;

$r_{i,j,b,d}$: required quantity of $\text{part}_{i,j}$ to make its immediate parent $\text{part}_{b,d}$, $(b, d) \in p_{i,j}$;

$t_{i,j}$: total processing time of $\text{part}_{i,j}$ to make its all immediate parent parts;

$t_{i,j,b,d}$: processing time of $\text{part}_{i,j}$ for parent $\text{part}_{b,d}$, $(b, d) \in p_{i,j}$;

$a_{i,j}$: ASRLT of $\text{part}_{i,j}$;

$n_{i,j,b,d}$: required quantity of $\text{part}_{i,j}$ to make unit end product from the leg that its immediate parent $\text{part}_{b,d}$ is on, $(b, d) \in p_{i,j}$;

$n_{i,j}$: required quantity of $\text{part}_{i,j}$ to make unit end product;

$s_{i,j} = \{(x, y)\}$: set of immediate children of $\text{part}_{i,j}$;

$\text{adu}_{i,j}$: average daily usage of $\text{part}_{i,j}$;

$c_{i,j}$: annual inventory cost of $\text{part}_{i,j}$;

$ai_{i,j}$: average inventory quantity of $\text{part}_{i,j}$;

$v_{i,j}$: unit price of $\text{part}_{i,j}$;

$\text{ltp}_{i,j}$: percentage usage of $\text{part}_{i,j}$ over ASR lead time;

$\text{vp}_{i,j}$: percentage of red zone base of $\text{part}_{i,j}$ that red zone safety accounts for;

h : annual inventory holding cost rate;

u : average daily usage of the end product;

Q : lead time for the end product requested by the customer (i.e., service time);

$y_{i,j}$: yellow zone quantity of $\text{part}_{i,j}$;

$g_{i,j}$: green zone quantity of $\text{part}_{i,j}$;

$\text{rb}_{i,j}$: red zone base quantity of $\text{part}_{i,j}$;

$\text{rs}_{i,j}$: red zone safety quantity of $\text{part}_{i,j}$;

(Decision variables)

$$X_{i,j} = \begin{cases} 1 & \text{if } \text{part}_{i,j} \text{ has inventory} \\ 0 & \text{if } \text{part}_{i,j} \text{ has no inventory.} \end{cases} \quad (1)$$

TABLE 1: Recommended impact ranges for green and red zone base.

Long lead time	20–40% usage over LT
Medium lead time	41–60% usage over LT
Short lead time	61–100% usage over LT

TABLE 2: Recommended impact ranges for red zone safety.

High variability	60–100% of red zone base
Medium variability	41–60% of red zone base
Low variability	20–40% of red zone base

4.1. *Formulation of SIP Problem.* The objective function in (2) is to minimize the total inventory cost for all items in the BOM. Equation (3) defines the annual inventory cost of $\text{part}_{i,j}$ by unit price multiplied by average inventory quantity and annual inventory holding cost rate:

$$\text{Min TC} = \sum_{\forall i} \sum_{\forall j} c_{i,j} \cdot X_{i,j}, \quad (2)$$

$$c_{i,j} = v_{i,j} \cdot ai_{i,j} \cdot h. \quad (3)$$

Ptak and Smith [1] propose that in the demand-driven materials requirement planning (DDMRP), the average inventory quantity can be calculated to be the total quantity of the red zone plus half quantity of the green zone, as (4). The yellow zone is usually set to be equal to the usage over ASR lead time, as (5):

$$ai_{i,j} = 0.5 \cdot g_{i,j} + \text{rb}_{i,j} + \text{rs}_{i,j} \quad (4)$$

$$y_{i,j} = \text{adu}_{i,j} \cdot a_{i,j}. \quad (5)$$

The lead time category of the buffer profile has a direct impact on the size of the green and red zone bases. Table 1 describes the recommended ranges of impact for the green and red zone bases for each lead time category. The minimum order quantity (MOQ) in this paper is not considered so that we can express the red zone base and green zone as (6) and (7), respectively:

$$\text{rb}_{i,j} = \text{ltp}_{i,j} \cdot y_{i,j}, \quad (6)$$

$$g_{i,j} = \text{ltp}_{i,j} \cdot y_{i,j}. \quad (7)$$

The variability category will size the red zone safety portion of the total red zone. Red zone safety is an expression of a percentage of the red zone base. Table 2 shows how red zone safety is sized. Red zone safety can be shown as

$$\text{rs}_{i,j} = \text{vp}_{i,j} \cdot \text{rb}_{i,j}. \quad (8)$$

In G-BOM, one part may have multiple immediate parent parts and is divided into several sections. We can get the number of requirements of each section to process one end product as (9). Then the number of requirements for one end product is the summation of all sections as (10). Average daily usage of $\text{part}_{i,j}$ is the average daily usage of end product

multiplied by required quantity of part_{*i,j*} for one end product, as in (11):

$$n_{i,j,b,d} = r_{i,j,b,d} \cdot n_{b,d}, \quad (b, d) \in p_{i,j}, \quad (9)$$

$$n_{i,j} = \sum_{(b,d) \in p_{i,j}} n_{i,j,b,d}, \quad (b, d) \in p_{i,j}, \quad (10)$$

$$\text{adu}_{i,j} = u \cdot n_{i,j}. \quad (11)$$

We know the processing time of a part to produce all its immediate parent part. The processing time for each section can be obtained according to the number of requirements to process one end product of each section portion of all requirements of this part to process one end product, as follows:

$$t_{i,j,b,d} = t_{i,j} \cdot \frac{n_{i,j,b,d}}{n_{i,j}}, \quad (b, d) \in p_{i,j}. \quad (12)$$

4.2. ASRLT for G-BOM. Turning the S-BOM to the G-BOM in this paper leads to more complicated expression of ASRLT. There can be more than one parent for one part in the G-BOM. In other words, at least two parts in the BOM have the common child part. In Figure 2, part 402 has two parent parts, 301 and 203. If there is no inventory holding for part 402, it has to be processed for its parent parts in make-to-order manufacturing. Among multiple parent parts of a common child part, which parent part to feed first becomes a scheduling problem. For N parents case, $N!$ different sequences need to be compared. In this paper, to make the logic easy to understand and simplify the expression of ASRLT, we consider the problem where each part has at most two parent parts.

Suppose that part_{*i,j*} and part_{*e,f*} are the two parent parts of part_{*k,l*}; that is, $s_{i,j} \cap s_{e,f} = (k, l)$. We assume that the common part is located in one lower level than that of the two parent parts whichever is lower; that is, $k = \max(i, e) + 1$.

Sequence 1. part_{*k,l*} is processed first to feed part_{*i,j*}. The ASRLT of part_{*i,j*} is maximum of two branches, as in the following:

$$a_{i,j} = \max_{(x,y) \in s_{i,j}} \begin{cases} (1 - X_{x,y}) a_{x,y} + t_{i,j} & \text{if } (x, y) \neq (k, l) \\ (1 - X_{x,y}) (a_{x,y} - t_{x,y,e,f}) + t_{i,j} & \text{if } (x, y) = (k, l), \end{cases} \quad (13)$$

$$a_{e,f} = \max_{(m,o) \in s_{e,f}} \{(1 - X_{m,o}) a_{m,o} + t_{e,f}\}. \quad (14)$$

According to (12), we get

$$t_{x,y,e,f} = t_{x,y} \cdot \frac{n_{x,y,e,f}}{n_{i,j}}. \quad (15)$$

We trace the indirect parent (which is the ‘‘ancestors’’ above the immediate parent in the BOM) of part_{*i,j*} and part_{*e,f*} along the ‘‘leg’’ in the BOM. Suppose that part_{*w,q*} and part_{*w,z*} are the parents of part_{*i,j*} and part_{*e,f*}, respectively. Since part_{*w,q*} and part_{*w,z*} are at the same level w , the two parts will have a common immediate parent part. Define $\varphi_1 = \max(a_{w,q}, a_{w,z})$.

Sequence 2. part_{*k,l*} is processed first to feed part_{*e,f*}:

$$a_{i,j} = \max_{(x,y) \in s_{i,j}} \{(1 - X_{x,y}) a_{x,y} + t_{i,j}\}, \quad (16)$$

$$a_{e,f} = \max_{(m,o) \in s_{e,f}} \begin{cases} (1 - X_{m,o}) a_{m,o} + t_{e,f} & \text{if } (m, o) \neq (k, l) \\ (1 - X_{m,o}) (a_{m,o} - t_{m,o,i,j}) + t_{e,f} & \text{if } (m, o) = (k, l). \end{cases}$$

Now we get the new $a_{w,q}$ and $a_{w,z}$. Define $\varphi_2 = \max(a_{w,q}, a_{w,z})$.

We choose Sequence 2 if $\varphi_1 \geq \varphi_2$, or Sequence 1, otherwise. We want to find a set $X^* = \{X_{i,j}\}$ which minimizes the total cost, given that the ASR lead time of the end product is no longer than the given service time Q . Now the SIP-G problem is mathematically formulated as follows:

$$\begin{aligned} \min \quad & \text{TC} = \sum_{\forall i} \sum_{\forall j} y_{i,j} \cdot h \cdot a_{i,j} \cdot X_{i,j} \\ \text{s.t.} \quad & a_{i,j} = (1.5 + \text{vp}_{i,j}) \cdot \text{ltp}_{i,j} \cdot \text{adu}_{i,j} \cdot a_{i,j} \\ & \text{adu}_{i,j} = u \cdot n_{i,j} \\ & n_{i,j} = \sum_{(b,d) \in p_{i,j}} n_{i,j,b,d} \quad (b, d) \in p_{i,j} \\ & t_{x,y,e,f} = t_{x,y} \cdot \frac{n_{x,y,e,f}}{n_{i,j}} \\ & a_{1,1} \leq Q \\ & a_{i,j} \begin{cases} \max_{(x,y) \in s_{i,j}} \begin{cases} (1 - X_{x,y}) a_{x,y} + t_{i,j} & \text{if } (x, y) \neq (k, l) \\ (1 - X_{x,y}) (a_{x,y} - t_{x,y,e,f}) + t_{i,j} & \text{if } (x, y) = (k, l) \end{cases} & \text{if } \varphi_1 < \varphi_2 \\ \max_{(x,y) \in s_{i,j}} \{(1 - X_{x,y}) a_{x,y} + t_{i,j}\} & \text{if } \varphi_1 \geq \varphi_2. \end{cases} \end{aligned} \quad (17)$$

TABLE 3: Parameter values for small instances in the G-BOM.

Part	1	2	3	4	5	6	7	8	9	10	11	12	13
t	2	5	7	6	4	4	3	3	3	4	3	7	11
r	1	1	1	3	2	5	1	1	1	(1,2)	2	4	6
p	(0,0)	(1,1)	(1,1)	(1,1)	(2,1)	(2,2)	(2,2)	(2,3)	(3,1)	(3,1) (2,3)	(3,2)	(3,2)	(4,2)
v	700	120	250	100	65	80	110	40	24	16	21	27	9
	Q = 10					$h = 0.25$					vp = 0.5		
ltp	Long lead time (26 + days)										30%		
	Medium lead time (11 to 25 days)										50%		
	Short lead time (1 to 10 days)										80%		

The formulation given in (17) is a nonlinear programming (NLP) model so that it is hard to solve. Therefore, a heuristic search algorithm is required to solve the model. In this study, a genetic algorithm is developed to solve the model. In the next section, we will present the genetic algorithm to efficiently solve the problem given in (17).

5. GA-Based Solution Procedure

5.1. Modeling. We solved the SIP-S problem using GA algorithm (Rim et al., 2014). In this paper, GA that combines the genetic operators such as selection, crossover, and mutation is also used to find the near-optimal solution for SIP-G problem. For the SIP-G problem, if there are n parts in the G-BOM, x more parts will exist in the new S-BOM after the BOM is split. So we define a chromosome as an array of $(n + x)$ binary genes instead of n in the previous paper, each of which also represents whether the corresponding part in the G-BOM holds inventory or not. To form the initial population of size m , chromosomes are randomly generated until m valid chromosomes are collected. A chromosome is a valid one only if two conditions are satisfied. One is that the ASR lead time of the end product is no longer than the lead time requested by the customer, as is in the SIP-S problem; the other is that all common parts in the S-BOM after splitting have to keep the same inventory status. For example, there are 13 parts in the G-BOM shown in Figure 2 and 15 parts in the S-BOMs in Figures 3 and 4. Parts 402(1) and 402(2) must have the same inventory status; and all of part 501P must have the same inventory status.

To evaluate the chromosome, we set the fitness value of the chromosome equal to a large enough number minus the total cost generated from the chromosome. Roulette wheel selection and elitism methods are applied to carry on the selection operation. In order to explore better solution, we use single-point crossover and mutation operations. In order to terminate the process of GA, termination conditions must be given. In this paper, we stop the algorithm after several generations, depending on the number of all solutions.

5.2. Computational Results. In order to demonstrate the application of the proposed methodology and to evaluate

TABLE 4: Parameter values for small instances in the S-BOM after splitting.

Part	BOM 1	Common data			BOM 2
	t_1	p	r	v	t_2
1	2	0	1	700	2
2	5	1	1	120	5
3	7	1	1	250	7
4	6	1	3	100	6
5	4	1	2	65	4
6	4	2	5	80	4
7	3	2	1	110	3
8	4	3	2	16	3
9	3	3	1	40	3
10	3	1	1	24	3
11	1	1	1	16	4
12	3	2	2	21	3
13	7	2	4	27	7
14	11	4	6	9	11
15	11	2	6	9	11

its performance, in this section two numerical examples are given.

5.2.1. Small Instances. There are 13 parts in the G-BOM given in Figure 2. Table 3 shows the parameter values. We assume that every part has the same variability percentage (vp) set as 50% for all parts. The lead time percentage (ltp) is 30%, 50%, and 80% for long, medium, and short lead time, respectively. The setting of the variability percentage and lead time percentage depends on the company environment. In much previous GA work, the crossover rate is typically set from 0.7 to 0.9; and the mutation rate is set from 0.05 to 0.15. We set the crossover rate P_c as 0.8; and the mutation rate P_m as 0.1. For small instances, we set the population size N as 80. The iteration will stop after 30 generations. As we mentioned, Figure 2 can be split to Figures 3 and 4, they have the same data of r , p , v shown in Table 4. In the S-BOM, p denote the number counted from the left of the immediate parent part, just the same as in the previous paper (e.g., $p_{2,2} = 1$, $p_{3,4} = 2$ in Figures 3 and 4).

TABLE 5: Experimental results.

P_c	P_m	N	Average ranking among 2,400
0.8	0.1	80	6

TABLE 6: Top 10 minimum total costs obtained from enumeration.

Ranking	TC
1	1990.6
2	2085.0
3	2089.0
4	2097.6
5	2122.6
6	2192.0
7	2196.0
8	2214.6
9	2217.0
10	2221.0

In order to show the effectiveness of the proposed GA to solve G-SIP problem, the result obtained from GA is compared with those from enumeration. We accept the solutions only when 402(1) and 402(2) are in the same inventory status, and two 501P have the same inventory status. The number of all solutions is $2 \cdot (2^{15} - 2^{11} \cdot 4) = 49,152$. However, 10 days of the ASRLT of the end product has to be met, so 2,400 feasible solutions are ranked from the minimum to maximum total cost generated from those feasible solutions. The number of feasible solutions increases as the lead time of the end product requested by the customer is longer. When we get the solution from GA, the ranking in the enumeration will be matched in the list of feasible solutions. Here, we run GA for 20 times and get the average ranking. The results are summarized in Table 5. Error $((TC_{GA} - TC_{min})/TC_{min})$ is not employed to test the effectiveness of the GA here (TC_{GA} represents the total cost of the best solution obtained from GA; TC_{min} is the minimum cost from enumeration), because the total cost of the feasible solution is not consecutive. Table 6 shows the top 10 minimum total costs obtained from the enumeration.

5.2.2. Medium-Sized Problem. In the previous section, we evaluate the effectiveness of GA to solve the G-SIP problem with small instances compared with the result obtained from enumeration. However, since GA is used to solve the larger problem, enumeration will take excessively long time. In this example, we have 50 parts in G-BOM, which means that there are at least 2^{50} solutions, the exact number of solutions depends on the number of parts that has multiple immediate parents and the number of parents. Table 7 lists the parameters for medium-sized problem in which the number of vp, lvp, h , P_c , and P_m is the same as those in small instances and Q is set as 16. For the same size problem, the computation time depends on the population size N and running generations that we set as the termination condition. We set to stop when the programming runs 200 generations. When N is selected as 80, the running time of the process is

TABLE 7: Parameter values for medium-sized problem.

Part	t	r	p	v
1	2	1	(0,0)	1200
2	5	1	(1,1)	900
3	7	1	(1,1)	890
4	6	1	(1,1)	910
5	4	2	(1,1)	860
6	4	3	(2,1)	880
7	3	2	(2,1)	800
8	3	1	(2,2)	780
9	3	2	(2,2)	790
10	4	3	(2,2)	750
11	3	2	(2,3)	690
12	5	3	(2,3)	660
13	5	4	(2,4)	600
14	4	3	(2,4)	640
15	2	1	(3,1)	700
16	4	3	(3,1)	660
17	4	(1,1)	(3,2), (3,3)	580
18	3	1	(3,3)	550
19	3	1	(3,4)	570
20	5	2	(3,4)	600
21	7	3	(3,6)	540
22	6	2	(3,6)	510
23	3	2	(3,7)	590
24	4	1	(3,8)	400
25	3	1	(3,9)	490
26	3	1	(3,9)	1200
27	5	2	(4,1)	900
28	4	2	(4,3)	890
29	3	1	(4,3)	910
30	2	3	(4,5)	860
31	5	1	(4,5)	880
32	4	1	(4,8)	800
33	3	3	(4,8)	780
34	3	2	(4,8)	790
35	10	(1,2)	(4,10), (4,11)	750
36	4	2	(4,12)	690
37	3	3	(4,12)	660
38	4	1	(5,1)	600
39	6	1	(5,1)	640
40	4	3	(5,2)	700
41	4	2	(5,4)	660
42	4	2	(5,4)	580
43	3	2	(5,4)	550
44	3	2	(5,6)	570
45	2	1	(5,9)	600
46	5	1	(5,9)	540
47	3	3	(5,11)	510
48	2	2	(6,2)	590
49	3	4	(6,5)	400
50	5	2	(6,8)	490

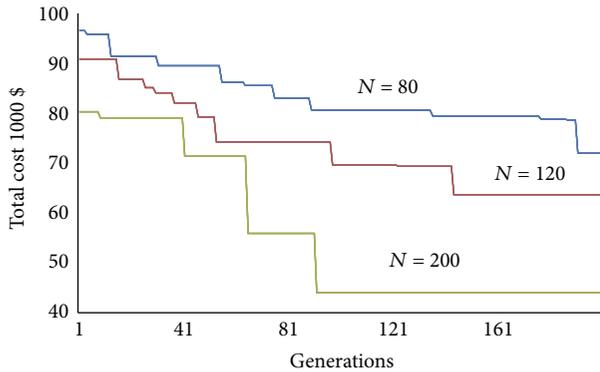


FIGURE 5: Convergence path for $N = 80, 120, 200$, respectively.

around 10 seconds in the PC. The computation time increases almost proportionally with the increased population size. When N is selected to be 80, 120, and 200, the convergence path graphs are presented in Figure 5, where the total cost converges faster and earlier for larger N .

In general, the larger population size (N) we use in genetic algorithm, the better solution we will obtain since more alternatives are considered in each generation as the result of crossover and mutation. That is why the solution quality is higher as N increases for the same number of generations in Figure 5. However, even if N keeps increasing, the solution quality will not keep improving as it converges to the optimum so fast. If N is set over 200, we will get the better solution with less number of generations; however, the total running time will increase. In the GA, we need to balance the solution quality with the computation time. So we need to take the appropriate N ; that is to say, it is not true that the bigger the N is, the better it is.

6. Conclusion

In this paper, we present a solution procedure for the strategic inventory positioning problem with general BOM, in which certain part has not only multiple child parts but also multiple parent parts. The paper concentrates on finding the optimal strategic inventory position to minimize the total inventory cost in the G-BOM, while satisfying a given ASR lead time of the end product. The G-BOM makes the calculation of ASRLT more complicated. The G-BOM can be split into S-BOMs as many as the factorial of the number of parents of an item in G-BOM, so computing the ASRLT becomes more complicated.

The practical contribution of this paper is that we provide a systematic solution procedure to determine the optimal location of stocks in the make-to-order manufacturing process in order to minimize the total inventory cost, while meeting the manufacturing lead time of the end product requested by the customer, when certain part has multiple parent parts in the BOM.

For further research related with current study, the limited capacity of the processes needs to be considered since common machines can be used in some stations; and due to

the limited capacity of the machine, some processing needs to be in a queue until other processing is finished.

Conflict of Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] C. Ptak and C. Smith, *Orlicky's Material Requirements Planning 3/E*, McGraw-Hill Education, New York, NY, USA, 2011.
- [2] S.-C. Rim, J. J. Jiang, and C. J. Lee, "Strategic inventory positioning for MTO manufacturing using ASR lead time," in *Logistics Operations, Supply Chain Management and Sustainability*, P. Golinska, Ed., EcoProduction, pp. 441–456, Springer, Basel, Switzerland, 2014.
- [3] D. C. Whybark and S. Yang, "Positioning inventory in distribution systems," *International Journal of Production Economics*, vol. 45, no. 1–3, pp. 271–278, 1996.
- [4] K. F. Simpson Jr., "In-process inventories," *Operations Research*, vol. 6, no. 6, pp. 863–873, 1958.
- [5] S. C. Graves and S. P. Willems, "Strategic safety stock placement in supply chains," in *Proceedings of the MSOM Conference*, Hanover, NH, USA, June 1996.
- [6] S. C. Graves and S. P. Willems, "Optimizing strategic safety stock placement in supply chains," *Manufacturing & Service Operations Management*, vol. 2, no. 1, pp. 68–83, 2000.
- [7] S. C. Graves and S. P. Willems, "Strategic inventory placement in supply chains: nonstationary demand," *Manufacturing & Service Operations Management*, vol. 10, no. 2, pp. 278–287, 2008.
- [8] E. Lesnaia, *Optimizing safety stock placement in general network supply chains [Ph.D. thesis]*, MIT Perations Research Center, Cambridge, Mass, USA, 2004.
- [9] T. L. Magnanti, Z. M. Shen, J. Shu, D. Simchi-Levi, and C.-P. Teo, "Inventory placement in acyclic supply chain networks," *Operations Research Letters*, vol. 34, no. 2, pp. 228–238, 2006.
- [10] P. Kaminsky and O. Kaya, "Inventory positioning, scheduling and lead-time quotation in supply chains," *International Journal of Production Economics*, vol. 114, no. 1, pp. 276–293, 2008.
- [11] K. Inderfurth, "Safety stock optimization in multi-stage inventory systems," *International Journal of Production Economics*, vol. 24, no. 1–2, pp. 103–113, 1991.
- [12] K. Inderfurth and S. Minner, "Safety stocks in multi-stage inventory systems under different service measures," *European Journal of Operational Research*, vol. 106, no. 1, pp. 57–73, 1998.
- [13] G. Ioannou, G. Prastacos, and G. Skintzi, "Inventory positioning in multiple product supply chains," *Annals of Operations Research*, vol. 126, no. 1, pp. 195–213, 2004.
- [14] W. Alderson, "Marketing efficiency and the principle of postponement," *Cost and Profit Outlook*, vol. 9, pp. 15–18, 1950.
- [15] L. P. Bucklin, "Postponement, speculation and the structure of distribution channels," *Journal of Marketing Research*, vol. 2, no. 1, pp. 26–31, 1965.
- [16] H. L. Lee, C. Billington, and B. Carter, "Hewlett-packard gains control of inventory and service through design for localization," *Interfaces*, vol. 23, no. 4, pp. 1–11, 1993.
- [17] T. Davis, "Effective supply chain management," *Sloan Management Review*, vol. 34, no. 4, pp. 35–46, 1993.

- [18] E. Feitzinger and H. L. Lee, "Mass customization at Hewlett-Packard: the power of postponement," *Harvard Business Review*, vol. 75, no. 1, pp. 116–121, 1997.
- [19] R. Ernst and B. Kamrad, "Evaluation of supply chain structures through modularization and postponement," *European Journal of Operational Research*, vol. 124, no. 3, pp. 495–510, 2000.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

