Heuristic Algorithm for the Safety Stock Placement Problem

Zied Jemai, Abderrahim Bendadou, Rim Kalai, Yacine Rekik

Heuristic algorithm for the safety stock placement problem

Bendadou Abderrahim[1], Kalai Rim[1], Jemai Zied[1,2], and Rekik Yacine[3]

[1] LR-OASIS, National Engineering School of Tunis,University of Tunis El Manar, Tunis, Tunisia
bendadou.abderrahim@gmail.com
[2] Université Paris-Saclay, CentraleSupélec, Laboratoire Genie Industriel, 3 rue Joliot-Curie 91190 Gif-sur-Yvette, France
[3] EMLYON Business School, 23 avenue Guy de Collongue, 69134 Ecully, Lyon, France.

Abstract. In this paper, we develop an iterative heuristic algorithm for the NP-hard optimization problem encountered when managing the stock under the Guaranteed Service Model in a multi-echelon supply chain to determine a good solution in a short time. Compared to the Baron solver that was restricted with a maximum time equal to 20000 seconds, we achieved on average more than 88% reduction in calculation time and about 0.3% cost reduction.

Keywords: Guaranteed Service Model, all or nothing property, algorithmic.

1 Introduction

Inventory optimization in the multi-echelon supply chain is achieved by allocating adequate safety stock at each stage to cover the uncertainty of customer demand. For this purpose, two models are proposed by researchers: The Stochastic Service Model (SSM) and the Guaranteed Service Model (GSM). The number of investigations dealing with GSM is large and has followed an increasing trend in recent years. To have a comprehensive view on GSM investigations, the reader is referred to the literature review in [1]. Papers dealing with methods for solving the GSM model optimization problem can be classified into two groups: those who worked on exact methods and others who focused on approximate methods. Simpson in [16] was the first to formalize the GSM for the multi-echelon supply chain. He argued that the optimal solution is none other than an extreme point of the polyhedron formed by the linear constraints of the program. This propriety entitled by “all or nothing ”. Several other investigations study the properties of the optimal solution in some specified cases such as type of the network, different service measures, we refer the reader to visit [2, 6–8, 13, 14, 17]. Particularly, the founded properties in some of these investigations enabled researches to develop exact resolution methods. [12] introduces dynamic programming algorithms to determine the optimal extreme point for systems with serial, assembly or distribution network. The investigation of [3], which is the first which extend the GSM model for general multi-echelon supply chains proposes solving the problem by adopting a dynamic programming algorithm. However, the proposed algorithm concerns only the systems with spanning-tree networks. [8] shows that the problem of finding optimal safety stock placement is NP-hard for general acyclic networks, Consequently, she develops a branch and bound algorithm to determine the optimal solution. She also improves the complexity of the algorithm presented by [3]. The authors in [4] develop a dynamic programming approach under the assumption of an arbitrary cost function that does not respect concavity, monotony, continuity, properties. They mainly concentrate on networks with clusters of communality, which are a particular case of general networks. In contrast, in our heuristic algorithm, the demand bound function still maintains its properties (concavity, monotonicity and continuity). The authors in [5] modified the dynamic programming algorithm of [3] for general acyclic network problem, they considered an arbitrary cost function that can be non-concave or non-monotone. [11] show that the safety stock placement problem can be formalized as a mixed integer programming (MIP) by approximating the concave cost function by a piecewise linear function. Then, they structure an iterative algorithm to solve the MIP problem. Resolution techniques based on exact methods require a high computation time and several researchers have attempted to reduce this time either by developing techniques for certain types of networks or by adopting approximate methods for general networks. Therefore, they propose heuristic algorithms that determine in a short time a suboptimal solution. In general, these calculation methods differ from each other because they are generally proposed for specific cases or they are based on specific insights. [15] provide for instance two heuristics algorithms based on the approximation of the objective function: the first one is based on the technique of iterative linear approximation while the second one employs the two-piece linear function as an approximation. Although they are fast, they often do not reach a good quality solution as in [11]. [13] proposed metaheuristic and heuristics methods (Linear Approximation , Simulated Annealing, Threshold Accepting, Tabu Search) to solve the optimization problem. [9] propose a heuristic algorithm based on genetic algorithm. In this study, we present an iterative heuristic algorithm based closely on the propriety of all or nothing for the safety stock placement problem when the supply chain is managed under the GSM model. We show efficiency numerically in terms of computational time and solution quality.

2 Model Description

2.1 Mathematical Formulation

In this section, we briefly discuss the generalization of GSM as introduced by [3] for supply chains with a complex network. To the best of our knowledge, this model applies to all types of networks.

Given a multi-echelon supply chain, let A, E be respectively the set of all existing arcs in the network and the set of all stages. We define by the following expression the maximum replenishment time M_j for stage j :

$$M_j = L_j + max\{M_i \setminus (i,j) \in A\} \tag{1}$$

Where L_j defines the lead time corresponding to stage j. Both the classical model and the generalized one share following assumptions:

- The lead times of all stages are deterministic.
- The lead times does not depend on the order size.
- External demand arrive stationary from i.i.d. process and occurs at stages facing the end customer.
- The inventory is controlled according to the periodic review base-stock policy with a common review period.
- Each stage is characterized by two times: the inbound service time SI_j which is the time to wait between placing an order and its receipt. The service time S_j that stage j offers to its customers. It is the time to wait between the arrival of an order from a customer and its satisfaction (The service times that stage j offers to its customers are assumed to be equal).

To face the demand variation, each stage j should have a stock during the net replenishment time $\tau_j \in \{1, 2, \ldots, M_j\}$ where

$$\tau_j = SI_j + L_j - S_j \tag{2}$$

Under the GSM model, each demand stage i should satisfy all the received orders during the net replenishment time which does not exceed the bound $D_i(\tau_i)$ where D_i is non decreasing concave function with $D(0) = 0$. To ensure 100% of service satisfaction, two conditions should be realized: Fist, each internal stage i should satisfy the internal order it receives from its immediate successors during the net replenishment time and which do not exceed the value $D_i(\tau_i)$. To do that, the order-up-to level B_i associated with stage i should be equal to the demand bound function at the value expressed by the net replenishment time related to stage i:

$$B_i = D_i(\tau_i) \tag{3}$$

Second, speed up the satisfaction of orders that exceed the bound by extraordinary measures such as (expedition, production overtime, outsourcing). We express the inventory level related to stage i at time t by :

$$I_i\left(t\right) = D_i\left(\tau_i\right) - \sum_{l=t-\tau_i+1}^{t} d_i(l) \tag{4}$$

Where $d_i(l)$ is the observed demand at time l at stage i. Referring to μ_i the mean of the demand coming to stage i for one period, we can express the expected safety stock at stage i as :

$$E\left[I_i\left(t\right)\right] = D_i\left(\tau_i\right) - \mu_i \tau_i \tag{5}$$

Let the unit holding cost at a stage i denoted by h_i, the GSM optimization aims to determine the optimal safety stock location and quantity that minimize the total expected holding cost. Such an optimization should satisfy the following constraints:

- Service times, inbound times, and replenishment times of all stages should be positive.
- All input items must be available before the beginning of the process, for this purpose, for each $(i, j) \in A$, we must assume $SI_j \geqslant S_i$.
- If DS defines the set of stages were the demand take place and SS defines the set of supply stages, $\forall i \in SS$ the inbound service time SI_i should be equal to the time offered by the external supplier which we denote $\underline{SI_i}$, likewise, $\forall i \in DS$, the service time S_i should be equal to time imposed by the end costumer which we denote $\overline{S_i}$.

Consequently, the GSM optimization problem is formulated as follows:

$$
\begin{array}{lllr}
 & \text{Min} & \sum_{i\in E} h_i \left(D_i(\tau_i) - \mu_i\tau_i\right) & \\
(\text{P}) & \text{subject to} & \tau_i = SI_i + L_i - S_i & \forall i \in E \\
 & & SI_i + L_i - S_i \geqslant 0, & \forall i \in E \\
 & & SI_j \geqslant S_i, & \forall (i,j) \in A \\
 & & SI_i = \underline{SI_i}, & \forall i \in SS \\
 & & S_i = \overline{S_i}, & \forall i \in DS \\
 & & SI_i \geqslant 0, & \forall i \in E
\end{array}
$$

As mentioned in the introduction, previous studies showed that the optimal solution of such an optimization problem is an extreme point of the feasible region. Besides, it is proved that problem (P) is NP-hard (Lesnaia [8]). We point out that the above mathematical formulation may be modified slightly when the supply chain is serial. In these systems, a single service time is considered for each stage, more precisely, given a serial system indexed from the most downstream to the most upstream which carries the index N. Each stage has only one upstream, So : $\forall i \in \{1, ..., N\} \ : SI_i = S_{i+1}$. Under this setting, the optimal solution remains expressed by the extreme point property as we have always a concave minimization under a polyhedron. Therefore, $\forall i \in \{1, ..., N\}$ the optimal service time at the stage i is expressed according to the following relation

$$S_i^* = S_{i+1}^* + L_i \ \mathit{or} \ 0$$

Researchers called this property all or nothing (store or not store), all if $S_i^* = 0$, nothing if $S_i^* = S_{i+1}^* + L_i$ (i.e. $\tau_i^* = 0$). This property is optimal when the system is serial. However, researchers did not talk about this property when the supply chain is with a complex network. We can extend this property for this kind of system by considering this setting, $\forall j \in E$

$$SI_j = \max\{S_i : \ (i, j) \in A\} \tag{6}$$

$$S_j = SI_j + L_j \text{ or } 0 \tag{7}$$

Under this setting, each stage either has the stock that cover the time related to the supplier that takes longer to respond to an order or it does not hold stock. We refer that all solutions inspired by this property are feasible. However, it may not be optimal when the network of the supply chain is complex.

3 Mathematical Results

In this section, we provide some mathematical results that our algorithms adopt. In the following, we remind the reader that when we use the term selected stage (or stages), we indicate that stage holds inventory according to all or nothing property (equations 6,7).
Let us consider a feasible solution inspired by all or nothing property in a multi-echelons supply chain with general network, we express this solution by the set of selected stages that we refer by $\mathcal{Y}$, on the other hand, for each stage i, we consider that $D_i - \mu_i \tau_i$ is non decreasing function. Let B be a subset of stages.

3.1 Notations

$c(i)$	Holding cost incurred at stage i by selecting stages in $\mathcal{Y}$,
$c_B(i)$	New holding cost incurred at stage i by selecting stages in $B \cup \mathcal{Y}$,
C	Total holding cost incurred at the system by selecting stages in $\mathcal{Y}$,
C_B	New total holding cost incurred at the system by selecting stages in $B \cup \mathcal{Y}$,
SI_i, S_i	Are respectively the inbound service time and service time at stage i by selecting stages in $\mathcal{Y}$,
SI_i^B, S_i^B	Are respectively the new inbound service time and service time at stage i by selecting stages in $B \cup \mathcal{Y}$.

Therefore, according to equations 6 and 7, if $v \notin B \cup \mathcal{Y}$ then $c(v) = 0$, else

$$c(v) = h_v(D_v(SI_v^B + L_v) - \mu_v(SI_v^B + L_v))$$

Let E_1, E_2 be two subsets of stages belonging to the same echelon (i.e. stages in $E_1 \cup E_2$ are located in the same echelon). We assume that $E_1 \cap \mathcal{Y} = \emptyset$ and $E_2 \cap \mathcal{Y} = \emptyset$. we have the following results :

Lemma 1. *If $E_1 \subseteq E_2$ then :*

- $\forall v \in E_1 \cup \mathcal{Y} \ : \ c_{E_2}(v) \leqslant c_{E_1}(v)$.
- $\forall v \in E_2 \setminus E_1 \ : \ c_{E_2}(v) > c_{E_1}(v)$
- $\forall v \in E \setminus E_2 \cup \mathcal{Y} \ : \ c_{E_2}(v) = c_{E_1}(v)$

Now, we assume that $E_1, E_2, \mathcal{Y}$ are two by two disjoint sets. We define E_3 by $E_3 = E_1 \cup E_2$

Lemma 2. *If* $C_{E_1} < C$ *then* $C_{E_3} < C_{E_2}$

On the other hand, we denote by E_4 the set of all stages belonging to the same echelon. In addition, we suppose that $E_4 \cap \mathcal{Y} = \emptyset$.
Let us consider the following procedure

Procedure 1.

- $V_1 = \emptyset$
- *Add to* V_1 *each stage* $v \in E_4 \setminus V_1$ *satisfying* $C_{V_1 \cup \{v\}} < C_{V_1}$

If $V_1 \neq \emptyset$ and V_2 is a subset of V_1 ($V_2 \subset V_1$) then

Lemma 3. $C_{V_1} < C_{V_2}$

Procedure 1 and lemma 3 constitutes a substrate idea on which we build the greedy algorithm. More precisely, at each echelon, if the stages are selected according to procedure 1, then we will be sure that no subset of the selected stages can be found under which the total cost is lower. Moreover, we can not select any other stages belonging to the same echelon, because this increases the total cost according to procedure 1.

3.2 The iterative heuristic algorithm

We reveal in this section the heuristic algorithm for the problem (P) based on all or nothing property. It aims to determine on a short time a good solution.
Given a multi-echelon supply chain composed of P echelons and N stages (the arcs between stages belonging to the same echelon are not supposed to be in the associated network). This algorithm requires the satisfaction of the following condition

$$\forall i \in DS : \overline{S}_i < L_i$$

Under this condition, we can know in advance that according to the optimal solution the demand stages will always be considered as stock points (i.e. $\forall i \in DS : \tau_i^* > 0$), and then we can easily build the initial feasible solution. Outside this condition, the algorithm may need an extension to be more efficient.
On the other hand. In order to apply this algorithm, stages must be numbered according to the following way :

- Number the stages belonging to echelon 1 from top to bottom, and continue to number stages belonging to the next echelons with the same way.

Let us consider new notations

F_j = set of all stages belong to echelon j.
O_n = set of selected stages at the iteration n.
e_{ij} = stage that belongs to echelon j and numbered by i.
n_j = number of the existing stages at echelon j.

As a summary, at each iteration, the algorithm apply procedure 1 from echelon 1 to the penultimate echelon. The algorithm stops only if the selected stages are identical to those of the previous iteration.

Algorithm 1: The iterative heuristic algorithm

```
Initialization :
1. Define the supply chain network.
2. Number the stages as the way described above.
3. Set O_0 = φ and O_1 = DS.
iteration n :
while O_n ≠ O_{n-1} do
    for j = 1 to P − 1 do
        O_n = O_n \ F_j;
        Let W_n be a different set of O_n;
        while O_n ≠ W_n do
            W_n = O_n;
            for i = Σ_{k=1}^{j-1} n_k + 1 to Σ_{k=1}^{j} n_k do
                if the total cost reduces by choosing the stage e_ij then
                    O_n = O_n ∪ e_ij;
```

4 Numerical Experiments

This section aims to test the effectiveness of the heuristic algorithm presented below. We code the algorithm and the optimization program by using Mathlab R2015b and AMPL respectively on a personal computer with Intel core i3-4005U processor (1.70 GHz) and 4 GR RAM. The baron solver (version 19.3.22) is adopted to compare the performance of our algorithm in terms of cost and computational time, since it is suitable for non-convex optimization problems [10]. We have set the Maxtime option of the baron solver to 20000 (s). In each test, we generate a random network with random lead times and standard deviations. More precisely, these parameters are chosen randomly in $[50, 150]$, and in $[3, 9]$ respectively. Regarding the holding cost parameters, we consider for each stage i that the value of h_i is equal to the maximum holding cost of its upstream stages plus a random scalar U. Furthermore, for the 60 generated problems, all the supply and customer service times are assumed to be zero while the safety factor is assumed to be equal to 2.
On the other hand, we consider four types of networks: serial, assembly, distribution and general acyclic. We do 15 tests for every type of network starting with a three-stage supply chain, then every time we increase the number of stages until 30.

Table 1 specifies the average, minimum, and maximum values of the CPU time observed by applying the iterative heuristic algorithm for each type of network. Moreover, it is shown in the CPU time gaps column the performance of its computation by comparing with the CPU time reached by the baron solver.
On the other hand, Table 2 shows the algorithm efficiency in terms of cost and

Network type	CPU Time (s)			CPU Time Gaps %		
	Mean	Min	Max	Mean	Min	Max
Serial	0.52	0.09	1.48	94.85	64.51	99.99
Assembly	1.03	0.1	2.52	80.03	41.44	96.00
Distribution	0.68	0.09	1.88	91.10	71.7	99.75
General acyclic	0.43	0.03	1.81	86.71	54.83	99.61

Table 1. The efficiency of the algorithm regarding the CPU Time

Network type	Cost Gaps %		
	Mean	Min	Max
Serial	0.74	0	4.72
Assembly	-0.47	-3.63	7.67
Distribution	0.83	0	8.14
General acyclic	0.24	-6	9.75

Table 2. The efficiency of the algorithm regarding the total cost of detention

quality solution by comparing it with the baron solver. Values with a negative sign mean that the solution obtained by the baron solver is better than that obtained by the algorithm, while those without any sign indicate that the solution obtained by the algorithm is better than the solution obtained by the baron solver.
Particularly, it is observed from table 1 that for each type of network the average CPU time is always less than $1.5s$ while the maximum value is always less than $3s$. On the other hand, through this algorithm we achieved more than 80% reduction in the CPU time.
Under the table 2, it is showed that the algorithm can reach good solutions with a reduction of more than 0.2% and more than 9% in best cases, whereas, under the cases where the baron solver access to good solutions we noted that all the gaps are less than or equal to 6%.

5 Conclusion

We introduce in this study an iterative heuristic algorithm for the NP-hard optimization problem that we face when managing the stock under the guaranteed service model for a multi-echelon supply chain. The algorithm's structure is based closely on the propriety of all or nothing which is not optimal for complex networks. This algorithm has proved its robustness through numerical experiments for any type of network, on the other hand. As a perspective to this work we propose to find an extension when $\underline{S}_i \geqslant L_i$ (i is a demand stage) or to apply it for general cyclic networks.

References

1. Eruguz, A.S., Sahin, E., Jemai, Z., Dallery, Y.: A comprehensive survey of guaranteed-service models for multi-echelon inventory optimization. International Journal of Production Economics **172**, 110 – 125 (2016)
2. Ghadimi, F., Aouam, T.: Planning capacity and safety stocks in a serial production–distribution system with multiple products. European Journal of Operational Research **289**(2), 533–552 (2021)
3. Graves, S.C., Willems, S.P.: Optimizing strategic safety stock placement in supply chains. Manufacturing & Service Operations Management **2**(1), 68–83 (2000)
4. Humair, S., Willems, S.P.: Optimizing strategic safety stock placement in supply chains with clusters of commonality. Operations Research **54**(4), 725–742 (2006)
5. Humair, S., Willems, S.P.: Technical note—optimizing strategic safety stock placement in general acyclic networks. Operations Research **59**(3), 781–787 (2011)
6. Inderfurth, K., Minner, S.: Safety stocks in multi-stage inventory systems under different service measures. European Journal of Operational Research **106**(1), 57 – 73 (1998)
7. Lee, K., Ozsen, L.: Tabu search heuristic for the network design model with lead time and safety stock considerations. Computers & Industrial Engineering **148**, 106717 (2020)
8. Lesnaia, E.: Optimizing safety stock placement in general network supply chains. Ph.D. thesis, Massachusetts Institute of Technology. Operations Research Center (2004)
9. Li, H., Jiang, D.: New model and heuristics for safety stock placement in general acyclic supply chain networks. Computers & Operations Research **39**(7), 1333 – 1344 (2012)
10. Lundell, A., Kronqvist, J.: Polyhedral approximation strategies for nonconvex mixed-integer nonlinear programming in shot. Journal of Global Optimization (2021)
11. Magnanti, T.L., Max Shen, Z.J., Shu, J., Simchi-Levi, D., Teo, C.P.: Inventory placement in acyclic supply chain networks. Operations Research Letters **34**(2), 228 – 238 (2006)
12. Minner, S.: Inventory placement in acyclic supply chain networks. Operations-Research-Spektrum **19**(4), 261 – 271 (1997)
13. Minner, S.: Strategic Safety Stocks in Supply Chains. Springer-Verlag Berlin Heidelberg (2000)
14. Minner, S.: Strategic safety stocks in reverse logistics supply chains. International Journal of Production Economics **71**(1), 417 – 428 (2001), tenth International Symposium on Inventories
15. Shu, J., Karimi, I.: Efficient heuristics for inventory placement in acyclic networks. Computers & Operations Research **36**(11), 2899 – 2904 (2009)
16. Simpson, K.F.: In-process inventories. Operations Research **6**(6), 863–873 (1958)
17. Yadegari, E., Alem-Tabriz, A., Zandieh, M.: A memetic algorithm with a novel neighborhood search and modified solution representation for closed-loop supply chain network design. Computers & Industrial Engineering **128**, 418–436 (2019)