Optimizing Safety Stock Placement in General Network Supply Chains

Stephen C. Graves, and Ekaterina Lesnaia

MIT

Abstract —**In the paper, we minimize the holding cost of the safety stock held in a supply chain modeled as a general network. By our assumption, the demand is bounded by a concave function. This fact allows us to formulate the problem as a deterministic optimization. We minimize a concave function over a discrete polyhedron. The main goal of the paper is to describe an algorithm to solve the problem without assuming any particular structure of the underlying supply chain. The algorithm is a branch and bound algorithm.**

Index Terms—**Base-stock policy; Multi-Stage Supply Chain Optimization; Safety Stock Placement; Branch and Bound Algorithm.**

I. Introduction

The problem being solved here deals with the amount of safety stock to hold at each stage of a supply chain. On one hand the amount must be such that a manufacturing company is able to serve its customers on time and satisfy most of the demand. On the other hand, the amount of stock to hold must be small to minimize holding and storage costs. By solving the problem a manufacturing company can protect itself against uncertain demand and provide a high level of service to its customers.

The problem was solved previously for a supply chain that can be represented as a spanning tree [1]. In the current paper, our objective is to find an algorithm to solve the problem for the general structure supply chains. The rest of the assumptions of the model remain the same as in [1]. In particular, we assume that each stage operates with a common review base-stock policy, the demand is bounded and the company wants to satisfy 100% of the customer demand. The formulation allows each stage of the supply chain to quote service times to the adjacent stages. The safety stock becomes a function of the service times. If we want to reduce the inventory at a particular stage of the chain, we set a longer service time, which allows the stage to store less and to delay production . Now, we have to find a right balance between quoting long service times at some stages and making the others wait and store inventory.

The assumption of bounded demand is the key assumption of the model. This assumption allows us to formulate the problem as a deterministic optimization. Moreover, we assume that the bounding function is concave. Therefore, the problem deals with minimization of a concave function over a set of constraints.

The algorithm developed here is a branch and bound algorithm. We show how the description of the optimal solutions helps us construct a branching tree. To develop the bounds on the branches, we use the algorithm for a spanning tree from [1].

In the next sections we first state the assumptions of the model and its mathematical formulation. Then, in section III we describe a subset of feasible solutions of the problem that contains an optimal solution. The algorithm is stated in section IV.

II. Assumptions and Formulation

A. Assumptions

Assumptions of the model were originally introduced in [1]. Here, we only outline the assumptions.

- **Multi-stage network.** We can model a supply chain as a network. Nodes and arcs of the network have natural interpretation in terms of the chain. Each node or stage in the network can be seen as a processing function in the chain. We place an arc from node *i* to node *j* if a product of stage *i* is needed in production at stage *j*. If a node is connected to several upstream nodes, then the node is an assembly requiring inputs from each of the upstream nodes. The nodes are potential locations for holding a safety-stock of the item processed at the node.

  Due to the interpretation of the network we assume that the network does not have directed cycles. This fact says that a component once processed in a node can not return back to the node in an assembly with other components.

November 2003. This work was supported in part by the Singapore-MIT Alliance and by the MIT Leaders for Manufacturing Program.

Stephen C. Graves is with the Sloan School of Management and the Engineering Systems Division at MIT, Cambridge MA 02139 USA (email: sgraves@mit.edu).

Ekaterina Lesnaia is a doctoral candidate in the MIT Operations Research Center, Cambridge MA 02139 USA (email: lesnaia@mit.edu).

Let N be the number of nodes and A be the set of arcs in the graph representing the chain.

- **Production lead-times.** We assume that each node j has a deterministic production lead-time T_j, where lead-time is the total time of production, given that all necessary components are available.

  Here we also introduce maximum replenishment time for a node j:

  $$M_j = T_j + \max\{M_j \mid (i,j) \in A\}$$

  The maximum replenishment time is the length of the longest directed path (with arc lengths T_j) in the network that terminates at node j, and represents the longest possible time to replenish the inventory at node j.
- **Demand process.** We assume that external demand occurs only in the demand nodes, namely in the nodes with zero out-degree. We denote the set of demand nodes as D. For each node j in D demand $d_j(t)$ comes from a stationary process with average demand per period μ_j.

  Any other node $i \notin D$ has only internal demand from its successors. We can calculate the demand in node i at time t by summing the orders placed by its immediate successors:

  $$d_i(t) = \sum_{(i,j)\in A} \theta_{ij} d_j(t),$$

  where a scalar θ_{ij} is associated with each node representing the number of units of upstream component i required per downstream unit j. From this relationship, we find the average demand rate for the node i to be

  $$\mu_i = \sum_{(i,j)\in A} \theta_{ij} \mu_j .$$

  The most important assumption of the model is that demand is bounded. In particular, for each node j there exists a function $D_j(F)$ for $F=1,2,\ldots,M_j$, such that
  1) for any period t
     $D_j(F) \geq d_j(t-F+1)+d_j(t-F+2)+\ldots+d_j(t)$;
  2) $D_j(0)=0$;
  3) the function is concave and increasing for $F=1,\ldots,M_j$;
  4) $D_j(F)-F\mu_j$ is increasing in F.
- **Base-stock replenishment policy.** All stages operate under a periodic-review base-stock policy with a common review period. We assume that there is no delay in ordering, therefore, all the nodes see customer demand once it occurs in the demand nodes. Based on the observed demand, inventory is replenished up to the bases stock level.
- **Guaranteed outbound service times.** We assume that node j provides 100% service and promises a guaranteed service time S_j to its downstream nodes. That means that demand $d_j(t)$ that arrives at time t must be filled at $t+S_j$. Note, we assume that for any non demand node j quotes the same service time to each of its downstream nodes i: $(j,i) \in A$.

  Also, we impose bounds on the service times for the demand nodes, i.e., $S_j \leq s_j$, $j \in D$, where $s_j \leq T_j$ is a given input that represents the maximum service time for the demand node j.
- **Guaranteed inbound service times.** Let SI_j be inbound service time for the node j. We define inbound service time to be the time for the node j to get all of inputs from nodes i: $(i,j) \in A$ and to commence production. We require that $SI_j \geq S_i$ for all arcs $(i,j) \in A$, since stage j cannot start production until all inputs have been received. We can show that, if the objective is to minimize the cost of the safety stock held in the chain, it is optimal to have [5].

  $$SI_j = \max_{(i,j)\in A} S_i .$$

All the parameters described here are known except for the service times. The service times are decision variables for the optimization.

B. Formulation

Suppose B_j is the base stock level for a node j and $I_j(t)$ is inventory in j at time t. Then the finished inventory at the stage j at the end of period t is

$$I_j(t)=B_j-d_j(t-SI_j-T_j,\ t-S_j),$$

where $d_j(a\ b)$ denotes demand at stage j over the time interval $(a,b]$.

To provide 100\% service level, we require $I_j(t) \geq 0$. To satisfy this requirement, we set the base stock $B_j=D_j(SI_j+T_j-S_j)$. Hence, the expected inventory at the stage j is

$$D_j(SI_j+T_j-S_j)-(SI_j+T_j-S_j)\mu_j,$$

which represents safety stock held at the stage j.

Now, we formulate the problem P of finding optimal guaranteed outbound service times S_j, j=1,…,N and inbound service times SI_j, j=1,…,N in order to minimize total cost of safety stock in the chain.

$$P \quad \min \quad \sum_{j=1}^{N} h_j \left\{ D_j\left(SI_j + T_j - S_j\right) - \left(SI_j + T_j - S_j\right)\mu_j \right\}$$

$$\begin{aligned} \text{s.t.} \quad & SI_j + T_j - S_j \geq 0, && j = 1,\ldots,N \\ & S_i \leq SI_j, && (i,j) \in A \\ & S_j \leq s_j, && j \in D \\ & S_j, SI_j \geq 0, && j = 1,\ldots,N \end{aligned}$$

This is a problem of minimizing a concave function over a polyhedron, which in general is NP-hard (see [2] and [3]). It is not proved whether problem P is or is not NP-hard. In the case the supply chain is presented as a bipartite graph a branch and bound algorithm can be used to find an optimal solution [5]. Here, we provide a branch and bound algorithm for a general network presentation of the chain.

III. NECESSARY CONDITIONS

In this section we describe solutions of the problem P that can potentially be optimal. To specify a set of such solutions, we introduce necessary conditions and describe feasible solutions that satisfy the conditions. The structure

of the solutions is critical in constructing the branch and bound algorithm in the following section.

We first define layers of the network. Node i belongs to the layer L_1 if there are no incoming arcs to the node i. Node i belongs to the layer L_k if there exists a directed path $(i_1,\ldots,i_k=i)$ of length k from a node $i_1 \in L_1$ to the node i and k is the length of the longest path of the type. Let the number of layers be K.

Lemma 1. *There always exists an optimal solution* $(S_1,\ldots,S_n,SI_1,\ldots,SI_n)$ *of the problem, such that all the inbound service times of the nodes from layer* L_1 *are 0*:

$$SI_j=0 \text{ for all } j \in L_1$$

and the remaining inbound service times are equal to the maximum of the outbound service times of the adjacent upstream nodes

$$SI_j=\max\{S_i,\ (i,j)\in A\} \text{ for all } j\in L_k,\ k>1.$$

Proof: We introduce a layer of dummy nodes L_0. The nodes are upstream of layer L_1 and are only connected to the nodes from L_1. For the purpose of the proof we add the new arcs to the set A. The nodes can be seen as an infinite source of raw materials for the production chain. The lead-times of the nodes are 0; consequently, we can assign outbound and inbound service times to be 0.

Suppose $\delta_j=SI_j-\max\{S_i,\ (i,j)\in A\}>0$ for $j\in L_k,\ k>0$. We define a new solution

$SI_j'=SI_j-\delta_j$

$S_j'=S_j-\min\{\delta_j,\ S_j\}$.

We can now consider two cases.

1. $\delta_j \le S_j$. Then the new solution is feasible and
$$SI_j'+T_j-S_j'=SI_j+T_j-S_j.$$
Therefore, the new solution has the same cost, is optimal and satisfies the lemma.
2. $\delta_j > S_j$. Then the new solution is feasible, but
$$SI_j'+T_j-S_j'=SI_j+T_j-S_j-(\delta_j-S_j)<SI_j+T_j-S$$
By the assumption 4) for the demand process, inventory at the stage j decreases as SI_j+T_j-S decreases, the cost of the new solution is strictly less than the cost of the optimal solution. Therefore, δ_j is always no greater than S_j.

The two cases show that we can always reduce the inbound service time to the maximal incoming inbound service time. In case of the nodes from the layer L_1, we can always reduce the inbound service times to 0, since the outbound service times of the nodes from L_0 are 0. **Q.E.D.**

Lemma 2. Let j be a demand node. In an optimal solution,

$$S_j=s_j.$$

Proof. The cost function decreases when outbound service times increase. Moreover, the only upper bounds on the outbound service times are the guaranteed service times s_i. We also assume that the guaranteed service times $s_i \le T_i$ for the demand nodes which prevents the arguments of the cost function from becoming negative. Therefore, the outbound service times are equal to the guaranteed service times. **Q.E.D.**

The two lemmas provide a characterization of optimal solutions for the problem. It is optimal to have service times for the demand nodes to be equal to the maximum guaranteed service times. It is also optimal for an inbound service time to be equal to the maximum outbound service time of its upstream nodes.

The results are rather intuitive. Postponing delivery of a product to the end customers till the latest possible moment gives greater flexibility in the earlier stages of the chain, and therefore more opportunities to minimize the total cost of the safety stock. The intuition behind the result of the first lemma might be as follows. In order to avoid unnecessary inventory in a node, the inbound service time of the node should be no greater than the largest guaranteed service time of its suppliers.

Observation 1. The objective function can be presented as a function of outbound service times S_i only. For each solution $(S_1,\ldots,S_n)$, the corresponding $(SI_1,\ldots,SI_n)$ can be reconstructed using lemma 1:

$$SI_j=0 \text{ for all } j\in L_1;$$
$$SI_j=\max\{S_i,\ (i,j)\in A\} \text{ for all } j\in L_k,\ k>1.$$

Observation 2. The objective function can be represented as a function of inbound service times SI_i only. For each solution $(SI_1,\ldots,SI_n)$, the corresponding $(S_1,\ldots,S_n)$ can be reconstructed using lemmas 1 and 2 and the fact that the inventory cost decreases when the outbound service time increases.

$$S_i=s_i \text{ for all demand nodes i;}$$
$$S_i=\min\{SI_j,\ T_i+SI_i;\ (i,j)\in A\} \text{ for all non demand nodes i.}$$

Observation 1 was previously used to construct a branch and bound algorithm for two-layer networks [5]. Observation 2 will now be used for a branch and bound algorithm for a general network.

Now, we formulate two lemmas that further characterize the optimal solutions of the problem. The objective of the lemmas is to provide a more detailed description of the extreme points of the polyhedron described by the constraints of the original problem P.

Lemma 3. If $i\in L_k$ and $k<K$, then in an optimal solution

$$S_i=0 \text{ or}$$

$S_i=SI_i+T_i$ or $S_i=SI_r+T_r$ for some r such that $(r,j)\in A$ and $j\in L_{k+1}$.

Proof. Suppose we have an optimal solution for the problem. Define set L to consist of nodes i such that $(i,j)\in A$ for some node $j\in L_k$, $k>1$. Let us renumber the nodes such that the nodes from L receive numbers 1 to m in the order of increasing outbound service times, i.e.,

$$S_1\le\ldots\le S_m,\ 1\ldots m\in L.$$

Suppose these outbound service times are optimal.

Suppose for some node $i\in L$ we know that $S_i<SI_j$ for all nodes j: $(i,j)\in A$. The only other condition on S_i is $S_i\le T_i+SI_i$. If $S_i<T_i+SI_i$, then we can increase S_i without violating any constraints, and decrease the value of the objective function. This contradicts the supposition of the optimality of $S_1,\ldots,S_m$. Hence, we can conclude that if $S_i<SI_j$ for all nodes j: $(i,j)\in A$, then $S_i=SI_i+T_i$.

Suppose now $S_i=SI_j=a>0$ for some node j: $(i,j)\in A$. Consider a subset of nodes C_a such that

- $i\in C_a$;
- $j\in C_a$ *if* $j\in L$ and $S_j=a$;
- $j\in C_a$ *if* $j\in L_k$ and $SI_j=a$.

Without loss of generality we can assume that C_a is connected. Let $u=\min\{j: j\in C_a\cap L\}$ and $v=\max\{j: j\in C_a\cap L\}$. That means

$$S_1\le\ldots<S_u=\ldots=S_v<\ldots\le S_m.$$

We note, that since $a=S_j$, $u\le j\le v$, we have $a\le SI_j+T_j$, $u\le j\le v$. Let $SI_{min}+T_{min}=\min\{SI_j+T_j: u\le j\le v\}$. We can express the total inventory function of the nodes contained in C_a as a concave function of a for feasible values of a. The parameter a is constrained below by S_{u-1}, and is constrained above by the minimum of S_{v+1} and $SI_{min}+T_{min}$. Thus, the function achieves its minimum at one of these end points.

If the minimum of the function were at $S_{u-1} = a$, then we should include node u-1 in C_a. The same is true if the minimum of the function were at $S_{v+1} = a$. Therefore, we can conclude that $a = SI_{min}+T_{min} < S_{v+1}$.

Suppose now, that $a=S_1$. Then we again define C_a. This time the total inventory function on C_a is concave in a and is defined on $[0,\min\{SI_j+T_j: 1\le j\le v\}]$. Therefore, the optimal a must be 0 or $\min\{SI_j+T_j: 1\le j\le v\}$. This completes the proof of the lemma. QED.

The following lemma is a direct consequence of the previous lemma and observation 2.

Lemma 4. If $j\in L_k$ and $k>1$, then in an optimal solution

$SI_j=0$ or

$SI_j= SI_i+T_i$ for some i such that $(i,r)\in A$ and $r\in L_k$.

We use lemma 4 to construct a branching tree for the algorithm described in the next section.

IV. ALGORITHM

As noted previously, the algorithm we describe here is a branch and bound algorithm [4]. In this section, we first specify the branching tree, and then the methods of constructing upper and lower bounds.

a) Branching tree

To construct the tree we number the nodes in the graph. We start by numbering nodes such that a node $i\in L_k$ only if there are no nodes with lower numbers in the layers with smaller k. In other words, we start by first numbering the nodes of layer L_1 and assign the numbers in any order to the nodes of the layer. The nodes of L_1 receive the numbers from 1 to the maximum number of nodes in the layer. Then we continue numbering the nodes of layer L_2 and so on.

Lemma 4 and the order of the nodes give us a way of constructing a branching tree. As we know from Observation 2, the objective function can be presented as a function of inbound service times only. By lemma 4, the inbound service times SI_j can take only a finite number of values in an optimal solution for all j. Therefore, the most natural step in constructing the branching tree is to try all possible values of SI_j. The ordering of the nodes provides a systematic way to do this.

We start from layer L_1. From Lemma 1 we know that SI_j for all nodes j in the layer take 0 value. Then we move to $L_2, L_3,\ldots,L_K$ in the order of increasing the node numbers.

Suppose we are at the layer L_k. The numbering of the nodes in the layer is $r, r+1, \ldots, p$. Starting from node r we let SI_r be 0 or SI_i+T_i where node i has a directed arc to a node in layer L_k. Then we do the next branching step for the nodes r+1 to p. Note here, once we assign a value of inbound service time for node r, we impose an upper bound on the outbound service times for all the upstream nodes that supply the node. If for some node i, we have assigned SI_j for all its successors m $(i,j)\in A$, then we can calculate the outbound service time S_i using lemma 3. In particular, the service time is $\min\{SI_i+T_i, SI_j\}$. The first term in the minimum is defined already, because node i belongs to a layer which we have already used in branching earlier in the process. Therefore, SI_i is already known. We have to include SI_i+T_i in the minimum to make sure that $SI_i+T_i-S_i\ge 0$ as imposed by the constraints of the formulation.

In general, by the time we branch on the nodes from layer k, we have already assigned a part of the solution to the nodes that are connected to the layers 2 to k-1 only. Let us call the set of such nodes N_k. Indeed, since the nodes themselves belong to the lower layers, we specify their inbound service times SI. Because they are connected only to the nodes from upstream layers, for which the inbound service times are set, we can calculate the outbound service times of the nodes from N_k as described earlier using lemma 3.

Example. As an example, let us consider the network presented on Fig.1. The lead times of the nodes of the corresponding supply chain are

$$T_1=1,\ T_2=2,\ T_3=3,\ T_4=4,\ T_5=5.$$

In figure 2, we show the tree for enumerating all of the solutions, which is the basis for the branch and bound algorithm. Note that as we generate the tree, the partial solution for the upstream nodes will impose some constraints on the inbound service times that need to be considered for the downstream nodes.

b) Lower bounds

As discussed in the previous section, a part of the solution can become known after each branching point, namely the nodes with known inbound and outbound service times. Effectively these nodes can be removed from the supply chain network. The nodes with known inbound service times, but unknown outbound service times stay in the network, but their outbound service times are bounded from above if inbound service times have been set for any downstream nodes that are direct descendents. The resulting subnetwork is a general network, and we want to develop a lower bound on the solution to the original problem restricted to the subnetwork.

To find a lower bound, we relax some of the constraints of the problem P. In particular, we remove some constraints of the form $S_i \le SI_j$ $(i,j)\in A$. Removing the constraints is equivalent to removing corresponding arcs from the graph. The goal is to remove the minimum number of arcs for which the resulting graph has a tree structure.

For the spanning tree we can apply the algorithm from [1] to solve the relaxed problem optimally and, therefore, to obtain a lower bound. The original algorithm from the

paper must be modified to accommodate possible bounds on some of both the outbound service times and the inbound service times of the nodes that remain in the graph.

c) Upper Bounds

Here, we show how to obtain an upper bound on the solution for the problem constrained to the subnetwork. It is enough to find any feasible solution. Given that we have a solution for the tree constructed for the lower bound, we can easily modify the solution to make it an upper bound.

The solution we obtained for the relaxed problem can be 'fixed' in the following way. The lower bound solution might not be feasible, since it can violate the constraints removed to obtain a spanning tree. The constraints are of the form $S_i \leq SI_j$ for $(i,j) \in$ A. From Lemma 1 we know that the inbound service time SI_j has to be equal to the maximum outbound service time of the upstream nodes directly connected to node j. Therefore, the solution can be fixed in two ways.

The first way to fix the solution is to increase the inbound service time of j to be equal to the maximum outbound service time of a directly connected node. The new inbound service time is

$$SI_j = \max S_i, (i,j) \in A.$$

The new solution is feasible. To see that, we check all of the constraints of problem $P.$. Indeed, the inbound service time stays non negative, since it can only increase. The constraint $SI_j + T_j - S_j \geq 0$ is also satisfied. And any constraint on the arcs $S_i \leq SI_j$ $(i,j) \in A$ is satisfied by construction of the new solution. Therefore, the new solution is feasible and is an upper bound on the optimal solution for the subnetwork.

Another way to fix the solution is to set the inbound service times to be the maximum of the inbound service times allowed by the constraints. In particular, the new solution will have

$$S_i = \min\{SI_i + T_i, SI_j\} \text{ for } (i,j) \in A.$$

The new solution is feasible. Indeed, $SI_i + T_i$ and SI_j are non negative. S_i becomes no greater than SI_j for all $(i,j) \in A$. Since S_i can only decrease, $SI_j + T_j - S_j$ can only increase and therefore can not become negative. Therefore, the new solution can be an upper bound on the solution for the subnetwork problem.

Another way of constructing an upper bound is to solve the problem optimally imposing some constraint on the unknown variables. For example, for the two-layer network, the problem can be solved optimally in polynomial time if we order the outbound service times of the component nodes [5].

V. Conclusions

In this paper we present a method to solve the problem of safety stock placement in supply chains modeled as a general network. We characterize the set that contains optimal solutions of the problem. Using this characterization, we show how to systematically enumerate the solutions, which is the basis for a branch and bound algorithm.

The performance of the branch and bound algorithm depends on the quality of the bounds. We continue to research how to improve these bounds. Whereas the calculation for the lower bounds seems reasonable, we expect we can improve upon the method for obtaining good upper bounds.

Another direction for the future research is to relax some of the assumptions of the model. For example, we are interested in developing a similar model that permits production capacity. This extension can make the model more useful for applications.

References

[1] S. Graves, and S.Willems, "Optimizing strategic safety stock placement in supply chains," *Manufacturing & Service Operations Management*, vol. 2 No. 1, pp. 68-83, Winter 2000.

[2] S.-J. Chung, "NP-completeness of the linear complementarily problem," *Journal of Optimization Theory and Applications*, 60:393-399, 1989

[3] O.L. Mangasarian, "Minimum support solutions of polyhedral concave problems," *Optimization,* 45 (1-4), pp. 149-162, 1999. Dedicated to the memory of of professor Karl-Heinz Elster.

[4] B. Korte, J.Vygen, *Combinatorial Optimization: Theory and Algorithms (Algorithms and Combinatorics, 21)*. Springer Verlag; 2nd edition, 2002.

[5] E. Lesnaia, "Optimizing safety stock placement in two-layer supply chains", *Proceedings of the 2003 SMA Conference*, Singapore, 5 pp., January 2003.

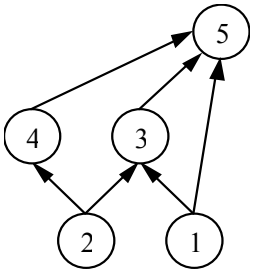

Fig. 1. Five-node supply chain example

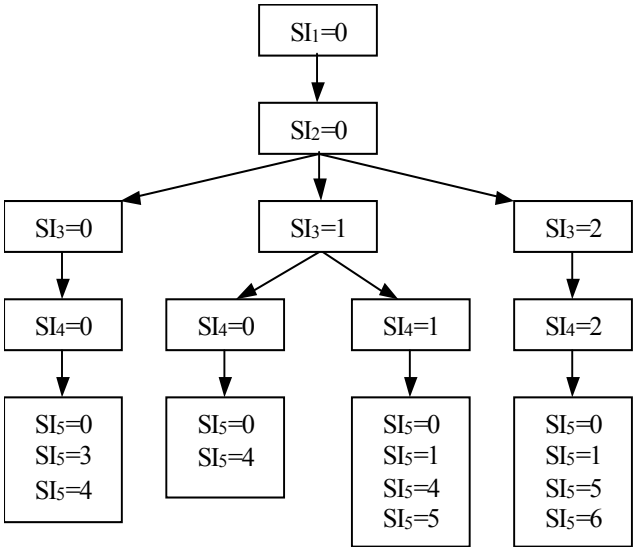

Fig.2. An enumeration tree for the network presented on Fig. 1.