# Progressive hedging as a meta-heuristic applied to stochastic lot-sizing

**Article** *in* European Journal of Operational Research · July 2001

DOI: 10.1016/S0377-2217(00)00116-8

# Progressive Hedging as a Meta-Heuristic Applied to Stochastic Lot-Sizing


Kjetil K. Haugen and Arne Løkketangen
Molde College
6400 Molde, Norway
and
David L. Woodruff
GSM, U.C. Davis
Davis CA 95616, USA

email:
Kjetil.Haugen@himolde.no
Arne.Lokketangen@himolde.no
DLWoodruff@ucdavis.edu



**Abstract**

In a great many situations, the data for optimization problems cannot be known with certainty and furthermore the decision process will take place in multiple time stages as the uncertainties are resolved. This gives rise to a need for stochastic programming methods that create solutions that are hedged against future uncertainty. The progressive hedging algorithm (PHA) of Rockafellar and Wets is a general method for stochastic programming. We cast the PHA in a meta-heuristic framework with the sub-problems generated for each scenario solved heuristically. Rather than using an approximate search algorithm for the exact problem as is typically the case in the metaheuristic literature, we use an algorithm for sub-problems that is exact in its usual context but serves as a heuristic for our meta-heuristic. Computational results reported for stochastic lot-sizing problems demonstrate that the method is effective.

# 1 Introduction

In a great many situations, the data for optimization problems cannot be known with certainty and furthermore the decision process will take place in multiple time stages as the uncertainties are resolved. This gives rise to a need for stochastic programming (SP) methods that create solutions that are hedged against future uncertainty. Stochastic programming methods produce solutions that are robust and also take into account the fact that there may be recourse in future decision periods.

When dealing with an optimization problem in the face of uncertainty, people often follow these steps:

1) Discuss the future and formulate a set of scenarios. Such scenarios tend to be few but capture the future spread to a certain extent. For instance, one may define an optimistic scenario (high future demand), a probable scenario (most likely future demand) and a pessimistic scenario (low future demand).

2) For each of these scenarios, a deterministic optimization is performed. In our setting, inventory holding and production schedules for each scenario would be found.

3) A comparison of the scenario-solutions is performed with the aim of finding a “blended” solution with a certain robustness towards future uncertainty. Surely, **one** solution is needed if decision making is the purpose of such a process. The solution used will – at least in principle – contain elements from the individual scenario-solutions hence, some kind of averaging procedure must be applied.

The progressive hedging algorithm (PHA) of Rockafellar and Wets [5] is a general method for stochastic programming. We cast the PHA in a meta-heuristic framework where the sub-problems that are generated for each scenario are solved heuristically. Thus, we make effective use of ideas from the meta-heuristics literature and bring them to bear on an important class of problems.

In the next section we describe the stochastic lot-sizing problem by giving it as an extension to the venerable deterministic lot-sizing problem of Wagner and Whitin. In §3 we sketch the PHA and indicate how it can be cast as a meta-heuristic for stochastic lot sizing problems. A number of important computational details are given in §4 along with computational results that demonstrate the effectiveness of the method. The paper closes with some conclusions and directions for further research.

## 2 A Stochastic Lot-Sizing Model

The static, deterministic lot-sizing model occupies a prominent place in the literature. The major initial contribution was made by Wagner and Whitin in the late fifties [10] and Eppen et al. [2] extended their results. The lot-sizing problem can be formulated in many ways, and we base our work on the following model.

$$\text{Minimize } Z = \sum_{t=1}^{T} \left[ c_t \delta(X_t) + h_t^+ I_t^+ + h_t^- I_t^- \right] \tag{1}$$

subject to:

$$X_t + I_{t-1}^+ - I_t^+ - I_{t-1}^- + I_t^- = d_t \quad \forall t \in \{1, \ldots, T\} \tag{2}$$

$$X_t \geq 0 \quad \forall t \in \{1, \ldots, T\} \tag{3}$$

$$I_t^+ \geq 0 \quad \forall t \in \{1, \ldots, T\} \tag{4}$$

$$I_t^- \geq 0 \quad \forall t \in \{1, \ldots, T\} \tag{5}$$

$$\delta(X_t) = \begin{cases} 0 \text{ if } X_t = 0 \\ 1 \text{ if } X_t > 0 \end{cases} \quad \forall t \in \{1, \ldots, T\} \tag{6}$$

where the decision variables are:

$\delta(X_t)$ = indicator of production in period $t$
$X_t$ = amount to be produced in period $t$
$I_t^+$ = ending positive inventory in period $t$
$I_t^-$ = ending negative inventory (backlogged) in period $t$

and the data is

$I_0$ = initial inventory
$c_t$ = setup cost in period $t$
$h_t^+$ = inventory holding cost in period $t$
$h_t^-$ = shortage cost in period $t$
$d_t$ = demand during period $t$

The problem formulated in (1) - (6) reflects the introduction of backlogging possibilities ($I_t^-$-variables) as well as shortage costs (the $c_t^- I_t^-$ term in the objective.) An alternative formulation involving the lost sales case may be obtained by replacing constraint (2) with

$$X_t + I^+_{t-1} - I^+_t + I^-_t = d_t \quad \forall t \in \{1, \dots, T\} \tag{7}$$

We stick to the backlogging-case and note that this model has been treated by Zangwill in a series of papers in the late sixties [13], [11], [12]. Another paper by Veinott [9] treats the same problem from a slightly different viewpoint. One of the key features of such models is that optimal solutions are characterized by the *Wagner-Whitin* property, which is that there is never both non-zero production and inventory in the same period. Zangwill and Veinott proved that this property holds for model (1) - (6); i.e., for optimal solutions

$$I^+_{t-1} X_t = 0 \tag{8}$$

$$X_t I^-_t = 0 \tag{9}$$

$$I^+_{t-1} I^-_t = 0. \tag{10}$$

This can be exploited to create a powerful, fast dynamic program (DP). Description of DP algorithms for the lot-sizing problem may be found in Veinott [9] and Zangwill [11].

A great deal of the current literature on lot-sizing concerns the dynamic lot-sizing problem (DLSP) where it is assumed that the problem data (particularly the demand data) is known for some of the periods and the remaining periods will be revealed later (see e.g., [6]). Our interest is the stochastic version (SLSP) where only the first period data is known and all other data is potentially stochastic.

We capture information about stochastic data by requiring the user to specify a reasonable number of representative *scenarios*. One way of visualizing a quite general time and state discrete stochastic process is by an event tree. Figure 1 shows an example with 2 states and 4 time stages. A distinct path from the root of the tree to a leaf is often referred to as a scenario - the grey shaded area in figure 1. Apart from the assumption of time and state discreteness, the event tree does not imply any other restrictions on the actual stochastic process. However, such a mechanism allows the user to specify events and probabilities in a natural way by conditioning on the events leading up to the current stage.

Each scenario gives a full set of random variable realizations and a corresponding probability. We index the scenario set, $\mathcal{S}$, by $s$ and refer to the probability of occurrence of $s$ (or, more accurately, a realization "near" scenario $s$) as $Pr(s)$. We then index all stochastic data by an $s$ to indicate its scenario. Our goal is to minimize the expected value of the objective

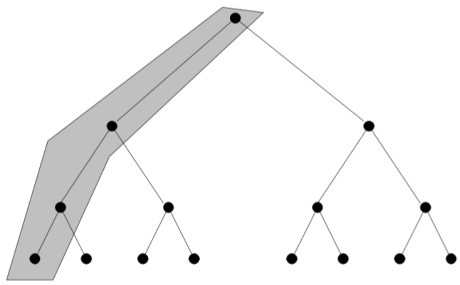

Figure 1: A simple event tree

function subject to a requirement that feasibility be maintained regardless of the scenario that is ultimately realized. We refer to solutions that satisfy constraints with probability one as *admissible*.

So, for example, if the demand data is stochastic, then we would seek to

$$\text{Minimize } \mathrm{E}[Z] = \sum_{s \in \mathcal{S}} Pr(s) \sum_{t=1}^{T} \left[ c_t \delta(X_{s,t}) + h_t^+ I_{s,t}^+ + h_t^- I_{s,t}^- \right] \tag{11}$$

subject to:

$$X_{s,t} + I_{s,t-1}^+ - I_{s,t}^+ - I_{s,t-1}^- + I_{s,t}^- = d_{s,t} \quad \forall s \in \mathcal{S},\ t \in \{1, \ldots, T\} \tag{12}$$

$$X_{s,t} \geq 0 \quad \forall s \in \mathcal{S},\ t \in \{1, \ldots, T\} \tag{13}$$

$$I_{s,t}^+ \geq 0 \quad \forall s \in \mathcal{S},\ t \in \{1, \ldots, T\} \tag{14}$$

$$I_{s,t}^- \geq 0 \quad \forall s \in \mathcal{S},\ t \in \{1, \ldots, T\} \tag{15}$$

$$\delta(X_{s,t}) = \begin{cases} 0 \text{ if } X_{s,t} = 0 \\ 1 \text{ if } X_{s,t} > 0 \end{cases} \quad \forall t \in \{1, \ldots, T\} \tag{16}$$

$$X, I, \delta \in \mathcal{N}_{\mathcal{S}} \quad . \tag{17}$$

The last constraint is added because although we must use a solution method that takes into account the fact that there will be future decision points, we cannot require the decision maker to know the future in order to know what to do at any point in time. We refer to a system of solution vectors as *implementable* if for scenario pairs $s$ and $s'$ that are indistinguishable up to time $t$, is true that $x_i(s, t') = x_i(s', t')$ for all $1 \leq t' \leq t$ and each vector element $i$. We refer to the set of all such solution systems as $\mathcal{N}_{\mathcal{S}}$ for a given set of scenarios, $\mathcal{S}$.

# 3 The Progressive Hedging Algorithm

Progressive hedging enforces the implementability constraint algorithmically. Due to space limitations, we cannot describe the algorithm in complete detail here. For more details, the reader is referred to Rockafellar and Wets [5] and Løkketangen and Woodruff [4]. In its general form, the PHA proceeds roughly as follows:

1. For each scenario $s$, approximate solutions are obtained for the problem of minimizing, subject to the constraints, $Z_s$. That is, the problem is solved for each scenario.

2. The values for an implementable solution are obtained by taking the mean of the solution values at each node in the scenario tree for the time index of the node for all scenarios at the node. These are used to update estimates for the dual variables associated with the (implicit) implementability constraints.

3. For each scenario $s$, approximate solutions are obtained for the problem of minimizing, subject to the constraints, the deterministic $Z_s$ plus terms that penalize lack of implementability. These terms strongly resemble those found when augmented Lagrangians are used.

4. If the solution has converged "sufficiently" then stop, otherwise goto 2.

The algorithm is intuitively appealing and has desirable theoretical properties: it converges to global optimum in the convex case, there is a linear convergence rate in the case of a linear stochastic problem, and if it converges in the non-convex case (and if the sub-problems are solved to local optimality) then it converges to a local optimal solution.

For the lot-sizing problem generated by step 1 in the PHA, the DP provides exact solutions. We use PHA as a meta-heuristic with the sub-problems generated for each scenario in step 3 solved heuristically. To understand these sub-problems we need to take a look at our formulation. We have four decision variables in our problem: $\delta(X_t), X_t, I_t^+, I_t^-$, at least from a modeling point of view. The PHA enforces implementability using a dual method so we need to assign a dual variable to each, which we do making use of $w_t^\delta, w_t^X, w_t^{I^+}, w_t^{I^-}$. Step 2 generates blended solutions, which we refer to as $\overline{\delta}, \overline{X}, \overline{I^+}, \overline{I^-}$. The optimization problems generated by step 3 are of a form that adds linear and quadratic terms to the objective. The following problem must be solved for every scenario, $s$, at every iteration of the PHA (but we omit the PHA iteration index in the interest of clarity):

$$\begin{aligned}
\text{Minimize } Z_s \quad = \quad & \sum_{t=1}^{T} c_t \delta(X_{s,t}) + h_t^+ I_{s,t}^+ + h_t^- I_{s,t}^- + \\
& w_t^\delta \delta_{s,t} + \frac{1}{2}\rho\left(\delta_{s,t} - \overline{\delta}\right)^2 + \\
& w_t^X X_{s,t} + \frac{1}{2}\rho\left(X_{s,t} - \overline{X}\right)^2 + \\
& w_{s,t}^{I^+} I_{s,t}^+ + \frac{1}{2}\rho\left(I_{s,t}^+ - \overline{I^+}\right)^2 + \\
& w_{s,t}^{I^-} I_{s,t}^- + \frac{1}{2}\rho\left(I_{s,t}^- - \overline{I^-}\right)^2 \qquad (18)
\end{aligned}$$

subject to:

$$X_{s,t} + I_{s,t-1}^+ - I_{s,t}^+ - I_{t-1}^- + I_{s,t}^- = d_{s,t} \quad \forall t \in \{1, \ldots, T\} \qquad (19)$$

$$X_{s,t} \geq 0 \quad \forall t \in \{1, \ldots, T\} \qquad (20)$$

$$I_{s,t}^+ \geq 0 \quad \forall t \in \{1, \ldots, T\} \qquad (21)$$

$$I_{s,t}^- \geq 0 \quad \forall t \in \{1, \ldots, T\} \qquad (22)$$

$$\delta(X_{s,t}) = \begin{cases} 0 \text{ if } X_{s,t} = 0 \\ 1 \text{ if } X_{s,t} > 0 \end{cases} \quad \forall t \in \{1, \ldots, T\} \qquad (23)$$

where $\rho$ is a parameter of the PHA.

The interesting point now is whether this problem behaves more or less as (1) - (6) when it comes to the possibility of using a DP. Let us look closer at this issue by returning to Wagner and Whitin's [10] original argument which may be described as follows: Suppose an optimal solution to the deterministic problem yields both bringing inventory into a period and setup in the same period (i.e. $I_t X_t > 0$). Then it is easy to accept that the resulting cost may be reduced by rescheduling the inventory such that all necessary demand is fulfilled by production in the given period. That is;

$$c_t \leq c_t + h_t^+ I_t^+ \qquad (24)$$

This reasoning leads to the opposite fact - namely; $X_t I_t = 0$. The condition is due to the binary structure of the set-up cost combined with the monotone increasing structure of the storage cost. It is sufficient to examine certain parts of the objective as described by equation (25);

$$\text{Minimize } Z = \sum_{t=1}^{T} c_t \delta(X_t) + h_t^+ I_t^+ + w_t^{I^+} I_t^+ + \frac{1}{2}\rho\left(I_t^+ - \overline{I^+}\right)^2 \ldots \qquad (25)$$

If we try to carry out the same type of argument based on the objective in equation (25), we run into problems. As $\rho > 0$, differentiating (25) with respect to $I_t^+$ twice yields;

$$\frac{\partial^2 Z}{\partial {I_t^+}^2} = \rho \qquad (26)$$

which indicates the strict convexity of this objective (in the $I_t^+$ variable). So we can not guarantee that a reschedule of inventory (i.e. decrease in $I_t^+$ and increase in $X_t$) will yield a total cost reduction. Hence, we can not prove the existence of an optimal strategy with characteristics as if $X_t I_t^+ = 0$. Unfortunately, this implies that PHA-sub-problem characteristics will not inherit those of the deterministic lot-sizing problems.

As noted by several authors (see for instance Kall and Wallace [3]), PHA does not need exact optimal solutions to sub-problems at each step. On the contrary, Kall and Wallace [3] state: *Not only* **can** *one solve scenario problems approximately, one* **should** *solve them approximately.* Rather than using an approximate search algorithm for the exact problem, we use an algorithm that would be exact were it not for the quadratic term. We make use of the DP algorithm that is based on planning horizon theorems (i.e. results of the Wagner/Whitin type as discussed earlier) even though we have established the fact that such an algorithm will not necessarily solve the sub-problems to optimality. This use of the DP for solving sub-problems is similar in spirit to work of Takriti et al [7] who used a DP to solve sub-problems for the unit commitment problem, although they use linear rather than quadratic sub-problems.

# 4 Computational Issues and Results

There is a severe shortage of published test cases for stochastic 0/1 MIPs in general, and none for the SLSP with scenarios. We have therefore built a test case generator that converts the stochastics inherent across the test cases in the paper by Trigeiro, Thomas and McClain [8] into stochastic optimization instances. This generator accepts as input a text file describing the base scenario together with stochastic and stage information. The output is in standard SMPS format [1]. The instances can be obtained on the web at www.gsm.ucdavis.edu/˜dlw/testcase-ejor.zip or by contacting the authors.

The stochasticity is in the form of a symmetrically distributed random demand for all stages after the first. This is represented in the form of 3 realizations of the demand at each stage, namely the average value, and the average value multiplied by 2/3 and 4/3, respectively. We randomly

| Stages | Scenarios | Variables | 0/1 Integers | Constraints |
|---|---|---|---|---|
| 3 | 9 | 52 | 13 | 26 |
| 4 | 27 | 160 | 40 | 80 |
| 5 | 81 | 484 | 121 | 242 |
| 6 | 243 | 1456 | 364 | 728 |

Table 1: Testcase and dimensions for the deterministic equivalent (DE)

generated test-problems for 3,4,5 and 6 stages. The corresponding number of scenarios are as shown in Table 1. Ten replicates were generated for each of the stage categories. All replicates for a given stage category have the same model of stochastic demand progression. The replicates differ in their costs, which were randomly generated.

Our stochastic programs have a corresponding deterministic equivalent (DE) which amounts to solving (11) - (17) directly. As our stochasticity is in the form of scenarios, this is a large, essentially block-diagonal mixed integer linear programming model (0/1 MIP). The dimensions of the resulting DEs are also shown in Table 1. We generated the DEs for all the test cases, and used them for comparisons.

Each scenario in the SP formulation corresponds to a subproblem for each iteration of the PHA. We solve these by the dynamic programming (DP) method described by Zangwill [11]. In the first PHA iteration there are no quadratic terms, so the DP finds the optimal sub-problem solutions. In the later stages, however, there is a quadratic penalty term that destroys the Wagner-Whitin property, as described in §3. The resulting non-optimal sub-problem solutions are then used as input to the hedging part of the PHA.

The PHA guarantees convergence only for convex problems. When applied to stochastic 0/1 MIPs convergence might be difficult to obtain. This is especially true if the 0/1 variables are indicator variables, where a change in the value might lead to large changes in the dependent real valued variables. The notion of *integer convergence* was introduced in Løkketangen and Woodruff [4]. Integer convergence is attained when all of the integer variables have converged, and not necessarily the other variables. The motivation for stopping the PHA before full convergence is achieved is that when the integers are fixed, then the values of rest of the variables are determined. This is done by forming the DE with all the integers fixed at their converged values. In our case it means that when we have decided in which periods we want to produce then the amount to produce is determined by solving the corresponding LP. The solutions obtained at integer convergence are all implementable.

The general solution method we have used is then as follows:

1. Run the PHA on the selected testcase.

2. Stop when integer convergence is reached.

3. Generate the corresponding DE(fixed) with the integers fixed at their converged values.

4. DE(fixed) is an LP and consequently can be solved relatively quickly using a commercial solver.

We also ran a series of tests limiting the PHA to one iteration, PHA(1), and solving the DE with those integers fixed that had converged, DE(partial). The motivation for this is that the integer variables that converge in the first PHA iteration might be considered to identify the "easy" part of the problem. On average 20% of the integers had converged, which results in a significant reduction in the time required for solution of the resulting DE(partial) as compared to the time for DE. It requires considerably more time than DE(fixed), but the benefit is an increase in solution quality.

We tested the algorithms using a computer with a 200MHz Pentium II processor and we used XPRESS-MP version 9 to solve DE, DE(partial) and DE(fixed). The results obtained when running our PHA algorithm on the test cases is shown in Table 2. Full convergence was not obtained, even when using a large number of PHA iterations and varying $\rho$ over a large set of values. Integer convergence was however easily obtained, needing less than three PHA iterations on average. A value of 0.8 for $\rho$ was used, but integer convergence was highly insensitive to this. The table shows the relative average quality and solution time between the PHA and DE and between PHA(1) and DE, respectively. The DEs where solved with a node-limit of 50000. This results in an actual average solution time for the 6-stage problem of 5550 seconds (compared to 11 seconds for the PHA). Proven optimality was only obtained for the 3-stage problems, while proven optimality takes more than 24 hours to achieve even for the 4-stage problems. As can be seen, solution quality is not degraded much by the PHA, and is much faster. If one contrasts the obtained quality with the accuracy with which the probability estimates of the stochastic variables in a practical situation might be estimated, the results are highly satisfactory. The total CPU time is always ordered DE, PHA(1), PHA. However, sometimes PHA and PHA(1) get better results as is indicated in Table 3. The tables do not show that for the 6-stage problems the PHA obtains better results in 3 of the 10 cases, and the

| | PHA/DE | | PHA(1)/DE | |
|---|---|---|---|---|
| Stages | Time | Quality | Time | Quality |
| 3 | 0.0497 | 1.05 | 0.177 | 1.04 |
| 4 | 0.0004 | 1.07 | 0.647 | 1.04 |
| 5 | 0.0008 | 1.05 | 0.886 | 1.03 |
| 6 | 0.0021 | 1.02 | 0.867 | 1.01 |

Table 2: Averages of the ratios of solution times and values

| | PHA/DE | | PHA(1)/DE | |
|---|---|---|---|---|
| Stages | Min | Max | Min | Max |
| 3 | 1 | 1.15 | 1 | 1.15 |
| 4 | 1 | 1.37 | 1 | 1.37 |
| 5 | 1 | 1.12 | 1 | 1.12 |
| 6 | 0.95 | 1.08 | 0.99 | 1.08 |

Table 3: Extreme values of the solution quality ratios

PHA(1) in 6 of the 10. The clear trend here is that as the number of stages increase, the relative solution quality of the PHA and PHA(1) gets better.

# 5 Conclusions and Directions for Further Research

The data for optimization problems are often uncertain, especially when based on future realizations. The associated decision process then progresses in stages, as the uncertainties are resolved. A decision maker needs to make a decision today, and needs solutions that hedge against these future uncertainties, including the possibilities of recourse.

These kinds of problems can be formulated as multistage stochastic programs. When the uncertainties are in the form of scenarios, the Progressive Hedging Algorithm can be applied. We have cast the PHA as a metaheuristic, using a subproblem solver that is exact in its usual context as a heuristic for our metaheuristic. The computational results show that high quality solutions can be obtained quickly, and clearly demonstrates the effectiveness of the method.

By casting the PHA in the metaheuristics framework, we are able to see that termination criteria can be varied to facilitate efficient cooperation with a branch and bound solver for partially fixed deterministic equivalents. We are also able to see that approximate solution of PHA sub-problems is clearly appropriate for heuristics. The concept of using the PHA as a metaheuristic can be extended to other problems by supplying different subproblem solvers.

# References


[1] J. R. Birge, M. A. Dempster, E. A. Gunn H. I. Gassmann, A. J. King, and S. W. Wallace. A standard input format for multiperiod stochastic linear program. *COAL (Math. Prog. Soc., Comm. on Algorithms) Newsletter*, 17:1–19, 1987.

[2] G. D. Eppen, F. J. Gould, and B. P. Pashihian. Extensions of the planning horizon theorem in the dynamic economic lot size model. *Management Science*, 15:268–277, 1969.

[3] P. Kall and S. W. Wallace. *Stochastic Programming*. Wiley, New York, 1994.

[4] A. Løkketangen and D. L. Woodruff. Progressive hedging and tabu search applied to mixed integer (0,1) multistage stochastic programming. *Journal of Heuristics*, 2:111–128, 1996.

[5] R. T. Rockafellar and R. J-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, pages 119–147, 1991.

[6] H. Stadtler. Improved rolling schedules for the dynamic single-level lot-sizing problem. Technical report, Technische Universität Darmstadt, Hochschulstr. 1, 1998.

[7] S. Takriti, J.R. Birge, and E. Long. Intelligent unified control of unit commitment and generation allocation. Technical Report 94-26, Univ. Michigan, Dept. of Ind. Eng. and Ops. Eng., 1995.

[8] W. W. Trigeiro, L. J. Thomas, and J. O. McClain. Capacitated lot sizing with setup times. *Management Science*, 35:3:353–366, 1989.

[9] A. F. Veinott. Minimum concave-cost solution of leontief substitution models of multi-facility inventory systems. *Operations Research*, 17:263–291, 1969.

[10] H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size formula. *Management Science*, 5:89–96, 1958.

[11] W. I. Zangwill. A deterministic multi-period production scheduling model with backlogging. *Management Science*, 13:105–119, 1966.

[12] W. I. Zangwill. Minimum concave cost flows in certain networks. *Management Science*, 14:429–450, 1968.

[13] W. I. Zangwill. A backlogging model and a multi-echelon model of a dynamic economic lot size production system - a network approach. *Management Science*, 16:507–527, 1969.