A Binary Multi-Objective Genetic Algorithm & PSO involving Supply Chain Inventory Optimization with Shortages, inflation

Dr. Kusum Gupta, Dr. Ajay Singh Yadav, Mr. Ankur Garg, Ms. Anupam Swami

1Banasthali University Jaipur, Rajasthan
2SRM University NCR Campus, GZB.
3MIET College Meerut, U.P.
4Govt. P. G. College Sambhal, U.P.

ABSTRACT

In this paper a supply chain inventory optimization model for shortages, inflation is considered under assumption that the inventory cost including holding cost shortages and inflation. The demand and holding cost, both are taken as variable. Shortages are allowed in the inventory model and a fraction of shortages backlogged at the next replenishment cycle. Transportation cost is taken to be negligible and goods are transported on the basis of bulk release pattern. A four level supply chain consists of a production, distributor Transportation and retailer, who are the cost bearers. It is necessary to have a coordinated approach between the tiers so that the chain is timed accurately for least inventory and minimum cost consequently maximum profits. In this paper, we consider a coordinated four echelon supply chain with a single production supplying a single type of product to single distributor and then to a single retailer. In this paper, we propose an efficient approach that effectively utilizes the Multi-Objective Genetic Algorithm and Multi-Objective Particle Swarm Optimization Algorithm for optimal inventory control. This paper reports a method based on Multi-Objective Genetic Algorithm and Multi-Objective Particle Swarm Optimization Algorithm to optimize inventory in supply chain management. We focus specifically on determining the most probable excess stock level and shortage level required for inventory optimization in the supply chain so that the total supply chain cost will be minimized. A numerical example is presented to illustrate the model and sensitivity is performed for a parameter keeping rest unchanged.

Keyword: Multi-Objective Genetic Algorithm, Swarm Optimization, echelon supply chain.

1.INTRODUCTION

Inventories are materials and supplies that a business or institution carry either for sale or to provide inputs or supplies to the production process. All businesses and institutions require inventories. Often they are a substantial part of total assets. An inventory is a stock of goods, which is held for the purpose of future production or sales. The manufacturing items in inventory are called stock keeping items, held at a stock (storage) point. Stock keeping items usually are raw materials, party finished items, finished (or prepared) goods, spare parts etc. Inventory levels for finished goods are a direct function of demand. Example which is based on inventory as follows: A hospital has method to control blood supplies and other important items as the blood stored in blood banks can only be used for a limited number of days. If this blood is overstocked for a long period cannot be used due to technological change and other reasons. This have to be disposed off as scrap material causing colossal waste of national resources. If the wastage of blood is reported in the newspaper it may act as a de-motivator to the voluntary donors. So, Inventory must be considered at each of the planning levels with production planning concerned with overall inventory, master planning with end items and materials requirements planning with components parts and raw material.

2.MULTI-OBJECTIVE GENETIC ALGORITHM

Discussions so far were limited to GA that handled the optimization of a single parameter. The optimization criteria are represented by fitness functions and are used to lead towards an acceptable solution. A typical single objective optimization problem is the TSP. There the sole optimization criterion is the cost of the tour undertaken by the salesperson and this cost is to be minimized. However, In real life we often face problem which require simultaneous optimization of several criteria. For example, in VLSI circuit design the critical parameters are chip area power consumption delay fault tolerance etc. While designing a VLSI circuit the designer may like to minimize area power consumption and delay while at the same time would like to maximize fault tolerance. The problem gets more complicated when the optimizing criteria are conflicting. For instance an attempt to design low-power VLSI circuit...
may affect its fault tolerance capacity adversely. Such problems are known as multi-objective optimization (MOO). Multi-objective optimization is the process of systematically and simultaneously optimizing a number of objective functions. Multiple objective problems usually have conflicting objectives which prevents simultaneous optimization of each objective. As GAs are population-based optimization processes they are inherently suited to solve MOO problem. However traditional GAs are to be customized to accommodate such problem. This is achieved by using specialized fitness functions as well as incorporating methods promoting solution diversity. Rest of this section presents the features of multi-objective GAs.

Multi-objective GA is designed by incorporating pareto-ranked niche count based fitness sharing into the traditional GA process.

This is presented as Procedure Multi-Objective-GA

Step 1:- Generate the initial population randomly.
Step 2:- Determine the pareto-optimal fronts $U_{p1}$, $U_{p2}$,........,$U_{pk}$.
Step 3:- If stopping criteria is satisfied then Return the pareto-optimal front $U_{p1}$
         Stop
Step 4:- for each solution $x_p$, evaluate the fitness.
Step 5:- Generate the mating pool $MP$ from population $P$ applying appropriate selection operator.
Step 6:- Apply crossover and mutation operations on the chromosomes of the mating pool to produce the next generation $P$ of population from $MP$.
Step 7:- Replace the old generation of population $P$ by the new generation of Population $P$
Step 8:- Go to step 2.

3. PARTICLE SWARM OPTIMIZATION

Particle swarm optimisation is initialized by a population of random solution and each potential solution is assigned a randomized velocity. The potential solutions called particles are then flown through the problem space. Each particle keeps track of its coordinates in the problem space which are associated with the best solution or fitness achieved so far the fitness value is also stored this value is called pbest. Another best value that is tracked by the global version of the PSO is the overall best value and its location obtained so far by any particle in the population. This value is termed gbest.

Thus at each time step the particle changes its velocity and moves towards its pbest and gbest this is the global version of PSO when in addition to pbest each particle keeps track of the best solution called nbest or lbest attained within a local topological neighbourhood of the particles the process is known as the local version of PSO

Multi-Objective Particle Swarm Optimization Algorithm

1: $P := 0$
2: $\{M_2, m_a, R_p, \nu, \nu\} = initialize()$
3: for $a := 1$: $U$
4: for $b := 1$: $X$
5: for $r := 1$: $R$
6: \( n_{X}^{(a+1)} = y_{n_{X}}^{a} + c_{1}d_{1}[V_{X} - m_{X}] + c_{2}d_{2}[U_{X} - m_{X}] \)
7: \( M_{X}^{(a+1)} = M_{X}^{a} + mN_{X} + \varepsilon \)
8: end
9: \( M_{X} := \) enforce Constraints (X)
10: \( Y_{X} := f(M_{X}) \)
11: if \( M_{X} \leq \varepsilon \vee e \in P \)
12: \( P := \{ e \in P / e < M_{X} \} \)
13: \( P := P \cup M_{X} \)
14: end
15: end
16: if \( M_{X} \leq V_{X} \vee (XM_{X} \leq V_{X} \wedge V_{X} < M_{X}) \)
17: \( V_{X} := M_{X} \)
18: end
19: \( U_{X} := \) selectGuide(X, A)
20: end

4. RELATED WORK


The notations used in the proposed model are shown as follows.
\[ a + bI(t) \] Demand rate in units per unit time (Normally one year)
\[ A_v \] Vendor’s ordering cost (\( \in \)/order)
\[ A_D \] Distributor’s ordering cost (\( \in \)/order)
\[ A_M \] Manufacturer’s setup cost (\( \in \)/setup)
\[ V_C \] Vendor’s unit cost (\( \in \)/unit)
\[ D_C \] Distributor’s unit cost (\( \in \)/unit)
\[ P_C \] Manufacturer’s unit cost (\( \in \)/unit)
\[ V_d \] Vendor’s ordering quantity in units
\[ D_q \] Distributor’s ordering quantity in units

\[ (D_q = (a + \gamma)R_q) \]
\[ (P_q = (\beta + \gamma)D_q) \]
\( (a + \gamma) \) The ratio of distributor’s replenishment quantity to Vendor’s replenishment quantity, appropriated to a positive integer.
\( (\beta + \gamma) \) The ratio of manufacturer’s replenishment quantity to distributor’s ordering quantity, appropriated to a positive integer.
\[ (h + \delta \gamma) \] holding charge or Interest rate in \( \in \)/Re/unit time
\[ v \] Vendor’s selling price (\( \in \)/unit)

\( \theta \) Inflation
\[ T_C \] Transportation Cost
\[ D \] Distance from Production’s to Vendor
\[ Q \] lot size per cycle (a decision variable)
\[ Z \] optimal backorder quantity
\[ \rho \] Linear backorder cost per unit
\[ S_C \] Shortages Cost
\[ T_{C_V} \] Total relevant cost of the Vendor (in \( \in \)) expressed in terms of \( R_q \)
\[ T_{C_D} \] Total relevant cost of the distributor (in \( \in \)) expressed in terms of \( (a + \gamma)R_q \)
\[ T_{C_P} \] Total relevant cost of the Production (in \( \in \)) expressed in terms of \( (a + \gamma)(\beta + \gamma)R_q \)
\[ T_{C_S} \] Total relevant cost of the supply chain (in \( \in \)) expressed in terms of \( (a + \gamma)(\beta + \gamma)R_q \)

\[ T_C \] Total Cost

Assumptions
The following features and assumptions are considered for the model.
1) Deterministic demand.
2) Instantaneous replenishment rate.
3) Distributor’s inventory is an integer multiple of retailer’s inventory.
4) Production’s inventory is an integer multiple of distributor’s inventory.
5) Shortages are allowed.
6) Transportation are allowed.
The annual total relevant cost of the retailer is given by the sum of annual ordering cost and holding cost at Vendor and it can be expressed as,

\[
TC_R (V_q) = \frac{A_V}{R_q} e^{ct} + \frac{V_q V_c (h + \delta t) e^{ct}}{2} \tag{1}
\]

The annual total relevant cost of the distributor is given by the sum of annual ordering cost and holding cost at distributor and it can be expressed as,

\[
TC_D [(\alpha + \gamma) V_q] = \frac{A_V (a + b I(t)) e^{ct}}{(\alpha + \gamma) V_q} + \frac{[(\alpha + \gamma) - 1] V_q D_2 (h + \delta t) e^{ct}}{2} \tag{2}
\]

The annual total relevant cost of the Production is given by the sum of annual ordering cost and holding cost at Production and it can be expressed as,

\[
TC_P [(\beta + \gamma) D_q] = \frac{A_M (a + b I(t)) e^{ct}}{(\beta + \gamma) D_q} + \frac{[(\beta + \gamma) - 1] D_q P_c (h + \delta t) e^{ct}}{2} \tag{3}
\]

\[
TC_P [(\beta + \gamma) V_q] = \frac{A_M (a + b I(t)) e^{ct}}{(\beta + \gamma) M_q} + \frac{[(\beta + \gamma) - 1] M_q M_c (h + \delta t) e^{ct}}{2} \tag{4}
\]

The annual total relevant cost of the supply chain is given by the sum of individual annual total relevant costs at retailer, distributor and Production and it can be expressed as,

\[
TC_S [(\alpha + \gamma) (\beta + \gamma) V_q] = \left[ A_V + \frac{A_D}{(\alpha + \gamma)} + \frac{A_M}{(\beta + \gamma)} \right] e^{ct} + \frac{A_V}{(\alpha + \gamma)} + \frac{A_M}{(\beta + \gamma)} \tag{5}
\]

\[
+ \left[ V_c + ((\alpha + \gamma) - 1) V_q + (\alpha + \gamma) [(\beta + \gamma) - 1] M_q \right] e^{ct} \frac{V_q V_c (h + \delta t)}{2}
\]

The annual total relevant cost of the transportation cost distance from Production’s to Vendor and it can be expressed as,

\[
TC_{(PC)} = \mu D \tag{6}
\]

The annual total relevant cost of the linear backorder cost and it can be expressed as,

\[
\text{SC} = \text{Linear backorder cost}
\]

\[
\frac{[a + b I(t)]}{Q} = \text{backorders}
\]

\[
\frac{Z}{Q} = \text{inventory cycles per year}
\]

Hence,

\[
\text{Number of backorders per Year} = \frac{[a + b I(t)] Z}{Q} \tag{7}
\]

If linear backorder cost per unit is \( p \) then

\[
\text{Linear backorder cost} = (\text{Number of backorders during the year}) \times (\text{Linear backorder cost per unit})
\]

\[
\text{Linear backorder cost} = \frac{[a + b I(t)] p Z}{Q} \tag{8}
\]

\[
TC \ (SC) = \frac{[a + b I(t)] p Z}{Q} \tag{9}
\]

The annual total relevant cost of the distributor is given by the sum of annual ordering cost and holding at distributor and it can be expressed as,

\[
\text{Total Cost} = \left[ TC_R [(\alpha + \gamma) (\beta + \gamma) V_q] + TC + SC \right] \tag{10}
\]

\[
\text{Total Cost} = \left[ \left[ A_V + \frac{A_D}{(\alpha + \gamma)} + \frac{A_M}{(\beta + \gamma)} \right] e^{ct} + \frac{V_c + ((\alpha + \gamma) - 1) V_q + (\alpha + \gamma) [(\beta + \gamma) - 1] M_q}{2} \right] \frac{V_q V_c (h + \delta t)}{2} + \mu D + \frac{[a + b I(t)] p Z}{Q} \tag{11}
\]

5. NUMERICAL ILLUSTRATION

The binary PSO algorithm presented for the incapacitated lot sizing problem is coded in C and run on an Intel P4 1.3 GHz, 128 equipment. The presentation of the binary PSO is measured with a traditional GA and the optimal Wagner and with in algorithm. For this purpose, a traditional GA is coded in C and test problems are generated arbitrarily for testing. The GA is a traditional one with a uniform crossover, simple inversion mutation and a tournament selection of size 2. We used the following parameters for the binary PSO and the traditional GA. For the binary PSO, the size of the population in the swarm is taken as the twice the number of periods. Social and cognitive parameters are taken as 11.5=ccc consistent with the literature. For the GA, the population size is the same as the binary PSO. The crossover and the mutation rates are 0.35 and 0.5 respectively. Both GA and PSO algorithms are run for 500 generations/iterations.
First test suit consisting of 5 problem instances with net requirements for 25 periods is generated from a uniform distribution, UNIF (25, 125), and the second test suit consisting of 5 problem instances with net requirements for 25 periods is generated from a uniform distribution, UNIF (50, 125). The total 10 problem instances are run for both GA and the binary PSO with holding cost of \( \sum_{k=1}^{n} \frac{h_k}{2} \) = € 0.25, ordering cost of \( A = € 50 \), shortages cost of \( SC = € 0.50 \) and Transportation Cost of \( T_{c} = € 0.75 \) in order to compare the results with those optimal by the Wagner-Whit in algorithm. For each problem instance, 10 replications are conducted. Minimum, maximum, average, and standard deviation are given together with the CPU times. As can be seen from Table 1 and 2 the binary PSO produced comparable results with GA, and it even produced better results. The GA was able to find the 7 optimal solutions out of 10 while the binary PSO was able find the 9 optimal solutions out of 10 even though the average standard deviation of the GA over 10 replications was slightly better than the binary PSO, i.e., \( = 3.47 \). In terms of the computational time, the GA took approximately 5 seconds for each instance while PSO took 8 seconds, which is computationally expensive than GA. But PSO’s good performance on finding optimal solutions more than GA compensates its computational inefficiency.

A Binary Particle Swarm optimization algorithm for lot Sizing Problem

### Table 1: PSO Results

<table>
<thead>
<tr>
<th>P</th>
<th>WW</th>
<th>PSO</th>
<th>OPT</th>
<th>BEST</th>
<th>MAX</th>
<th>AVG</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2105.50</td>
<td>2105.50</td>
<td>2111.75</td>
<td>2108.00</td>
<td>3.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2177.00</td>
<td>2177.00</td>
<td>2185.75</td>
<td>2178.75</td>
<td>2.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2051.00</td>
<td>2051.00</td>
<td>2051.00</td>
<td>2051.00</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2039.00</td>
<td>2039.00</td>
<td>2039.00</td>
<td>2039.00</td>
<td>2.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2011.25</td>
<td>2011.25</td>
<td>2022.50</td>
<td>2022.70</td>
<td>1.58</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2124.50</td>
<td>2124.50</td>
<td>2134.60</td>
<td>2124.70</td>
<td>2.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2034.50</td>
<td>2034.50</td>
<td>2034.70</td>
<td>2068.30</td>
<td>3.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2150.50</td>
<td>2150.50</td>
<td>2150.00</td>
<td>2140.00</td>
<td>0.48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2216.50</td>
<td>2216.50</td>
<td>2206.50</td>
<td>2216.50</td>
<td>0.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2012.00</td>
<td>2012.00</td>
<td>2013.00</td>
<td>2014.00</td>
<td>3.50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: GA Results

<table>
<thead>
<tr>
<th>P</th>
<th>WW</th>
<th>GA</th>
<th>OPT</th>
<th>BEST</th>
<th>MAX</th>
<th>AVG</th>
<th>STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2205.50</td>
<td>2205.50</td>
<td>2211.75</td>
<td>2208.00</td>
<td>4.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>2277.00</td>
<td>2277.00</td>
<td>2285.75</td>
<td>2378.75</td>
<td>5.69</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2151.00</td>
<td>2151.00</td>
<td>2151.00</td>
<td>2251.00</td>
<td>2.50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2139.00</td>
<td>2139.00</td>
<td>2139.00</td>
<td>2139.00</td>
<td>4.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2111.25</td>
<td>2111.25</td>
<td>2122.50</td>
<td>2222.70</td>
<td>2.78</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2224.50</td>
<td>2224.50</td>
<td>2234.60</td>
<td>2524.70</td>
<td>4.11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2134.50</td>
<td>2134.50</td>
<td>2234.70</td>
<td>2168.30</td>
<td>4.85</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2250.50</td>
<td>2250.50</td>
<td>2250.00</td>
<td>2440.00</td>
<td>1.48</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>2316.50</td>
<td>2316.50</td>
<td>2306.50</td>
<td>2316.50</td>
<td>2.35</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>2112.00</td>
<td>2112.00</td>
<td>2113.00</td>
<td>2114.00</td>
<td>6.50</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 6. CONCLUSIONS

This paper gives a comparative study along a line of research investigating optimal inventory decisions in a four level supply chain with single Production, single distributor supplying a single type of product to a single vendor. The comparison is based on the algorithm adopted to solve the four echelon problem illustrated by a numerical example. The model compares the optimized total relevant cost at the Production, distributor and vendor. It is observed that the vendor’s replenishment quantity is more in case of particle swarm optimization method along with similar number of shipments as shown by values of positive integers. In addition, it is noted that genetic algorithm gives a lesser output for the total relevant cost of total chain as compared to particle swarm optimization. Hence, it may be concluded from the comparative study that genetic algorithm generates better optimal values for decision variables and objective functions. The scope of the project is cited in industrial applications as optimization of inventory storage in industries.

### REFERENCES


