International Journal of Management Science
Vol 13, No 2, November 2007

Multi-Item Inventory Problems Revisited Using Genetic Algorithm*

Prasun Das**
SQC & OR Unit, ISI, Kolkata, India

(Received Aug. 2006; Revised May 2007; Accepted May 2007)

ABSTRACT

This paper makes an attempt to compare the two important methods for finding solutions of multi-item inventory problem with more than one conflicting objectives. Panda et al. [9] discusses a distance-based method to find the best possible compromise solution with variation of priority under the given weight structure. In this paper, the problem in [9] is revisited through the Pareto-optimal front of genetic algorithm with the help of a situation of retail stocking of FMCG business. The advantages of using the solutions from the perspective of the decision maker obtained through multi-objective optimization are highlighted in terms of population search, weighted goals and priority structure, cost, set of compromise solutions along with prevention of stock-out situation.

Key words : Genetic Algorithm, Multi-objective Optimization, Pareto-optimality, Priority Structure, Stock-out Prevention

1. Introduction

A supply chain management system typically works on the "pull" system philosophy, where the orders are placed to the warehouse (or, stores) from various retail outlets, based on the sale movement and thereby generating demands. In order to have a control on the supply chain so that the demands of the customers can be serviced without increasing the capital tie-up in inventory significantly, an effective system to facilitate smooth operation of the supply chain is required. However, from business

* The author would like to thank the significant contributions of the reviewers for providing valuable comments and fruitful suggestions to improve the paper.

** Corresponding author, E-mail: dasprasun@rediffmail.com

perspective, fulfilling 100% demand is infeasible due to the restriction in maximum capacity of warehouse and a constraint over total inventory cost. Further, maintaining a low inventory even at a lesser cost might be a risky proposition from the stock-out situation. The priority of some important items also needs to be taken care at times while stocking. Keeping in mind these ever conflicting business goals, it is necessary to formulate a multi-objective optimization problem to achieve implementable solutions from business perspective.

One approach could have been to opt for the classical method of multi-objective goal programming to obtain the best possible solution. But, one problem with the approach would be to arrive at only one solution to present to the decision-maker for implementation, as the various classical multi-objective optimization methods deal with point-search technique. If the given solution cannot be implemented due to some other constraint or unstated restriction, the decision-maker would have no other choice, but to reconstruct the problem in some other method. Several optimization techniques to deal with such problems have been proposed to consider even the decision makers' priorities over various goals. Still, the proposed techniques are mainly of point-search type, from which it is difficult to obtain several acceptable solutions, so that decision maker may choose one of them according to his own will. Recently, Panda *et al.* [9] discussed a method in which instead of determining priority structure the best possible compromise solution is obtained using ideal point distances from the different solutions with variation of priority under the given weight structure of the problem. This paper has several weaknesses like.

a) The D_1-distances used by the authors of [9], to find the best compromise solution against the ideal solution, does not take into consideration the cost aspect whereas the main objective of the case described in this paper was to minimization of total cost. As a result, the total cost related to the best compromise solution was found more than the total cost for the ideal solution. Even, the total cost for some other priority structure, ignoring the distance measure D_1, is less than the cost for best compromise solution, without violating the constraint on minimum capacity.

b) The decision maker would have no choice to select from multiple solutions based on the goal structure and the priority levels to make decision-making process flexible. In [9], the number of solutions obtained from non-linear goal programming method (NLGP) using the distance measure D_1 is small (six, in number), along with violation of priority structure as suggested by the decision-maker.

c) Goal programming, in case [9], does not consider the stock-out situation while minimizing the cost and determining economic order quantity.

To overcome this, it is decided to utilize the concept of the genetic algorithm (GA) where optimization can be done using population search technique using the similar database from [9] and conceptualized a situation of retail stocking of footwear business from the perspective of the decision maker.

This paper makes the following contributions to the literature. First, we explain the major weaknesses of the paper by Panda et al. [9]. Second, we compare the findings of [9] using genetic algorithm and discuss the utilities of Pareto-optimal front to the decision-maker for selecting the multiple solutions depending on the goals and priority structure set by him/her. Third, we highlight the unnoticed features in [9] against the solutions obtained through GA in terms of cost, quantity, goals and priority structure, including prevention of stock-out situation.

This paper is organized as follows. In Section 2, basic concepts and advantages of genetic algorithms for multi-objective optimization problems are described. In Section 3, we present the required definitions and the notations used in this paper. The optimization problems, under certain constraints, to be solved using GA are formulated in Section 4. In Section 5, the solutions obtained using goal programming and non-dominated sorting genetic algorithm along with Pareto-optimal front is presented. To mention specifically, certain drawbacks in results obtained through goal programming approach and in comparison to advantages of GA solutions to the decision maker are elaborated in Section 5.3. Section 6 recommends the solution from the point of view of the decision-maker and closes the discussion with the highlighting features of the study.

2. Materials and Methods

In classical approach, one problem in case of multi-objective optimization would be to arrive at only one solution to present to the decision-maker for implementation, as the various classical multi-objective optimization methods deal with point-search technique. Goal programming (GP) technique, developed by Charnes and Cooper [3] was first used in the field of multi-item inventory control in a supply-chain management system. A number of papers and books have been published showing the formulations using linear and non-linear goal programming, such as, Panda *et al.* [9], Basu *et al.* [2], Charnes and Collomb [4] and Charnes *et al.* [5]. In case of expansion or merger of store-based distribution channels, a stochastic model is used to develop a near-optimal distribution (ordering and allocation) policy with minimum total expected distribution cost by Tang *et al.* [1], where the demands at sales locations are stochastic and correlated among sales locations, and the depots hold no inventory. If the given solution cannot be implemented due to some other constraints or unstated restrictions, the decision-maker would have no other choice, but to reconstruct the problem in some other method. Hence, it is decided to utilize the concept of the genetic algorithm (GA) where optimization can be done using population search technique.

GA's are adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic. The basic concept of GA's is designed to simulate processes in natural system necessary for evolution, specifically those that follow the principles first laid down by Charles Darwin for survival of the fittest. As such they represent an intelligent exploitation of a random search within a defined search space to solve a problem. Not only does GA's provide an alternative method to solving problem, it consistently outperforms other traditional methods in most of the problems link. Many of the real world problems involved finding optimal parameters, which might prove difficult for traditional methods but ideal for GA's, [7]. Recently, GAs have been applied in different areas like neural network, traveling salesman, scheduling, numerical optimization and pattern recognition etc. Mondal and Maiti [8] have discussed an approach to solve NLP problems by developing multi-item fuzzy EOQ models using GA.

Genetic algorithms, differing from conventional search techniques, start with an initial set of random solutions, called *population*. Each individual in the population is called a *chromosome*, which is a string of symbols, representing a solution to the problem at hand. The chromosomes evolve through successive iterations, called *generations*. GA's, unlike traditional optimization techniques, use payoff information (fitness function) instead of derivatives or other auxiliary knowledge. During each generation, the chromosomes are evaluated using some fitness function and fitter chromosomes are selected for *reproduction*. For the next generation new chromosomes, called *offspring*, are formed by either crossing over two chromosomes in a mating pool, or mutating a chromosome using probabilistic transition rules.

Among the various species, non-dominated sorting version of GA (NSGA-II, [6]) is used in this case of multi-objective optimization problem, where the solution is obtained as a Pareto-optimal front; thereby providing the decision maker a number of alternate solutions to the conflicting objectives and helping him to make a choice of his own. Key concepts of this algorithm are defined as follows;

Concept of domination:

A solution $x^{(1)}$ is said to dominate the other solution $x^{(2)}$, if both the following conditions are true:

1. The solution $x^{(1)}$ is no worse than $x^{(2)}$ in *all* objectives.
2. The solution $x^{(1)}$ is *strictly better* than $x^{(2)}$ in *at least one* objective.

Non-dominated Set:

Among a set of solutions P, the non-dominated set of solutions P′ are those that are not dominated by any member of the set P.

Pareto-optimality:

The non-dominated set of the entire feasible search space S is the *globally* Pareto-optimal set or simply Pareto-optimal set.

3. Definitions and Notations:

We now consider the general model for multi-item inventory problem, as discussed in [9], and then discuss the concept of ideal solution, compromise solution and the weight structures when priority based non-linear goal programming technique is adopted to solve the problem.

3.1 General model for multi-item inventory problem

We first consider the objective function as a sum of *m* (number of items) independent cost functions, where each of them can be defined as goal under goal programming methodology and a target level for each goal can be set. The entire objective function related to cost of multi-items is to be minimized subject to certain restrictions (constraints) imposed on the problem. Accordingly, the general model for this programming problem can be formulated as follows.

$$\text{Find } q^* = (q_1^*, q_2^*, \cdots, q_m^*) \text{ so as to}$$

$$\textit{minimize } \sum_{i=1}^{m}\left(C_i q_i + \frac{C_{si}}{q_i}\right)$$

$$\textit{subject to } \sum a_{ij} q_i \le = \ge b_j,\ 1 \le j \le n \tag{1}$$

$$\textit{and } 0 < q_i \le Q_i,\ 1 \le i \le m$$

where $C_1, C_2, \cdots, C_m$ are the per unit ordering costs, $C_{s1}, C_{s2}, \cdots, C_{sm}$ are the set-up costs and $Q_1, Q_2, \cdots, Q_m$ are the inventory level of $q_1, q_2, \cdots, q_m$ (decision variables) respectively. The coefficients a_{ij}s could be proportional allocation of items to satisfy the requirements (supply or demand) b_j for j^{th} location.

3.2 Ideal Solution

While formulating the problem (1) for solving using goal programming, deviational variables are introduced to allow possible variations from the stated goals. Further, preemptive priorities are assigned to goals that force the most important goals to be satisfied before lesser goals are considered.

Now, suppose, the set $\{q_1^{(l)}, q_2^{(l)}, \cdots, q_m^{(l)}\}, 1 \le l \le p!$ contains a total of $p!$ solutions, which are achievable by considering the $p!$ combinations of p priority levels. Now, in this case, since, the problem is a cost minimization problem, a solution $q^* = (q_1^*, q_2^*, \cdots, q_m^*)$ is said to be the ***Ideal Solution*** for some l, $1 \le l \le p!$, where

$$\left(C_i q_i^* + \frac{C_{si}}{q_i^*}\right) \le \left(C_i q_i^{(l)} + \frac{C_{si}}{q_i^{(l)}}\right) \quad \forall\, i = 1, 2, \cdots, m.$$

3.3 Compromise Solution

In practical situations, when a particular priority structure under the imposed weight structure (see *Sec. 3.4*) by the decision-maker is not obtained, the most appropriate priority structure is identified through a solution nearer to the ideal solution. This solution is called the *best compromise solution,* which is obtained by minimizing the sum of all the deviational variables associated with the decision variables and the priority levels.

3.4 Weight Structure

Weights are assigned with the deviational variables, primarily based on the choice of the decision maker, to include the goals in the model. In this work, we consider the same weight structures of [9] as designed (or, imposed) by the decision-maker.

- *1st Weight Structure : Special attention to four items* – I-0101, I-0104, I-0202, I-0301 (see Table-1 in Section 4)
  $WS^1_p : (3d^+_{1p} + 2d^+_{4p} + 2d^+_{6p} + d^+_{8p})$
- *2nd Weight Structure : Cost goals (excepting the special four items)*
  $WS^2_p : (2d^+_{2p} + d^+_{3p} + 3d^+_{5p} + 3d^+_{7p} + d^+_{9p} + d^+_{10p} + 2d^+_{11p} + d^+_{12p})$
- *3rd Weight Structure : Item and Financial restriction*
  $WS^3_p : (d^-_{13p} + d^-_{14p})$

The suffix p represents the level of priority ($p = 1, 2, 3$).

The authors of [9] determine *the best compromise solution* instead of assigning the priorities to the goals. Thus to ensure that all the goals have the equal opportunity to acquire the highest priority, the solutions of six problems corresponding to $p! = 3! = 6$ priority structures are displayed in Section 5.1, along with the *ideal solution*.

3.5 Ideal Point Distance (IPD)

It is defined as the sum of the absolute deviation of all the solutions $\{q_1^{(l)}, q_2^{(l)}, \cdots, q_m^{(l)}\}$, $1 \le l \le p!$ from the ideal solution $\{q_1^*, q_2^*, \cdots, q_m^*\}$. Mathematically,

$$(IP_D)^l = \sum_{i=1}^{m} | q_i^* - q_i^{(l)} |$$

$$\text{and, } IP_{D_{opt}} = \min_{1 \le l \le p!} (IP_D)^l = (IP_D)^k,\ 1 \le k \le p!$$

Hence, $\{q_1^{(p)}, q_2^{(p)}, \cdots, q_m^{(p)}\}$ is the *best compromise solution* according to the problem situation [9].

4. A Practical Situation

We consider the similar database of [9] to revisit the solutions of the problem using genetic algorithm. The practical consideration in using this database can be described as follows.

The retail stores of a footwear company, with a maximum capacity of 460 units, maintain stock of different groups (Gents, Ladies, Kids, Hawai etc.) of footwear on monthly rental basis. The three main groups and the selected top class footwear sub-category are considered here. The instantaneous capacity of the stores should be a minimum 80% of total capacity. The budget limitation for a retail of the company is 0.9 million dollar. Simultaneously four footwear sub-categories from three groups need special attention while stocking. The objective of the company is to minimize the inventory cost at the retail stores without inviting the stock-out condition. The necessary cost components, the respective inventory levels and the investment required are given in Table 1.

Table 1. Database for the problem

Item Group (coded)	Item Sub-category (coded)	Ordering Cost/unit	Set-up cost	Inventory Level	Average investment/unit	Total Cost
	(q_i)	(C_i)	(C_{si})	(Q_i)		
IG-01	I-0101	500	27600	80	550	38000
	I-0102	600	25000	40	650	20000
	I-0103	700	30000	20	750	14000
	I-0104	400	26000	30	460	10000
IG-02	I-0201	1400	68000	60	1600	80000
	I-0202	1700	69000	50	1950	86000
	I-0203	1200	45000	40	1400	48000
IG-03	I-0301	4000	62000	40	4500	160000
	I-0302	6500	103000	30	7500	200000
	I-0303	12700	138000	20	14000	260000
	I-0304	900	40000	30	1200	25000
	I-0305	1600	61000	20	1800	20000

4.1 Problem Formulation for GA

We now formulate the problem stated in Section 3 as a multi-objective optimization problem for solving using GA. We make use of the data given in Table 1 and formulate the multiple objective functions first that are conflicting in nature. As stated in the problem, the stock out condition at the stores must be discouraged while minimizing the cost of inventory; therefore, it immediately comes into mind the conflicting situation about maximizing the quantity of each item along with minimizing their total cost of inventory.

The multi-objective optimization problem can now be formulated as follows.

Find $q^* = (q_1^*, q_2^*, \cdots, q_{12}^*)$ so as to

maximize Total Item : $q_1 + q_2 + \cdots + q_{12}$

minimize Total Cost : $55q_1 + 65q_2 + 75q_3 + 46q_4 + 160q_5 + 195q_6 + 140q_7 + 450q_8 + 750q_9 + 1400q_{10} + 120q_{11} + 180q_{12}$

So, the problem involves two objective functions, conflicting in nature

The constraints, formulated in line with the goal programming problem formulation in [9], are as follows.

4.1.1 Constraints

Cost related

Since, for each item, a fixed amount of cost is set as target (maximum possible), the equivalent item-wise cost restriction in GA formulation can be written as

$$5q_1 + \frac{276}{q_1} \le 380; \quad 6q_2 + \frac{150}{q_2} \le 200; \quad 7q_3 + \frac{300}{q_3} \le 140; \quad 4q_4 + \frac{260}{q_4} \le 100; \quad 14q_5 + \frac{680}{q_5} \le 800;$$
$$17q_6 + \frac{690}{q_6} \le 860; \quad 12q_7 + \frac{450}{q_7} \le 480; \quad 40q_8 + \frac{620}{q_8} \le 1600; \quad 65q_9 + \frac{1030}{q_9} \le 2000;$$
$$127q_{10} + \frac{1380}{q_{10}} \le 2600; \quad 9q_{11} + \frac{400}{q_{11}} \le 250; \quad 16q_{12} + \frac{610}{q_{12}} \le 200$$

Note that, since the above cost for each item is independent to each other, the problem of total cost minimization will not be affected.

Item related

Since, the lower bound of the capacity of the warehouse is 80% of the total capacity of 460 units, the equivalent constraint in GA for item restriction can be written as

$$q_1 + q_2 + \cdots + q_{12} \geq 368$$

Financial restriction

The equivalent form for financial restriction in GA formulation can be written as

$$55q_1 + 65q_2 + 75q_3 + 46q_4 + 160q_5 + 195q_6 + 140q_7 + 450q_8 + 750q_9 + 1400q_{10} + 120q_{11} + 180q_{12} \leq 90000$$

4.1.2 Limits of Decision variables

Since the NSGA-II algorithm for GA requires both sided bounds for the decision variables to generate the search space using the genetic operators, the upper bounds for this problem are considered as the maximum level of inventory for each item. The lower bounds are obviously set with non-negative restrictions.

$$0 \leq q_1 \leq 80;\ 0 \leq q_2 \leq 40;\ 0 \leq q_3 \leq 20;\ 0 \leq q_4 \leq 30;\ 0 \leq q_5 \leq 60;\ 0 \leq q_6 \leq 50;$$
$$0 \leq q_7 \leq 40;\ 0 \leq q_8 \leq 40;\ 0 \leq q_9 \leq 30;\ 0 \leq q_{10} \leq 20;\ 0 \leq q_{11} \leq 30;\ 0 \leq q_{12} \leq 20.$$

5. Solutions

5.1 Optimization using Goal Programming

The authors in [9] provide solutions for the non-linear programming problem considering six priority structures, based on penalty function method [5]. The ideal solution, as reported by considering the cost to be minimized, is given by

$\left\{q_1^{(\min)}, q_2^{(\min)}, \cdots, q_{12}^{(\min)}\right\}$ = {75.19, 33.08, 12.20, 20.66, 54.00, 43.81, 34.11, 34.13, 22.01, 13.07, 22.17, 9.71}. We now calculate the ideal point distances IP_D of all the six solutions and represent below.

Table 2. Calculation of IP_Ds

l (run)	$\lvert q_i^* - q_i^{(l)} \rvert$												IP_D
1	0.85	0.21	0	1.13	1.97	1.03	0.15	0.16	0	0.63	0	0	**6.13**
2	0.93	0.32	0.02	1.22	2.06	1.13	0.25	0.21	2.72	1.15	0.79	0.03	10.83
3	2.49	3.36	3.07	5.07	2.24	3.61	2.75	2.59	3.94	2.61	4	1.91	37.64
4	2.5	3.16	2.87	4.87	2.04	3.71	2.55	2.39	3.74	2.4	3.8	1.71	35.74
5	0	0	0.68	0	0	0	0	0	1.65	0.65	0.78	0.12	**3.88**
6	0.55	0.9	1.11	0.63	0.66	0.67	0.79	0.79	2.35	0.95	1.54	0.37	11.31

From Table 2 it is found that the minimum of the IP_D from the ideal solution is 3.88. This corresponds to the priority structure $P_1P_2P_3 = WS^3{}_1WS^1{}_2WS^2{}_3$ and the best compromise solution is $\{q_1^{(5)}, q_2^{(5)}, \cdots, q_{12}^{(5)}\}$ = {75.19, 33.08, 12.88, 20.66, 54.00, 43.81, 34.11, 34.13, 23.66, 13.72, 22.95, 9.83}. The total cost corresponding to this solution is 0.8699526 million dollar, whereas, with respect to ideal solution, the total cost becomes 0.8468156 million dollar.

Incidentally, in this work, it is found that for the first (l = 1) priority structure ($P_1P_2P_3 = WS^1{}_1WS^2{}_2WS^3{}_3$) where the next minimum IP_D is 6.13, the sum of total items becomes 380.27, which is feasible. Further, the total cost corresponding to this solution is 0.8628499 million dollar, which is even less than the cost of *best compromise solution*. We call this solution as *next best compromise solution* and the solution is $\{q_1^{(1)}, q_2^{(1)}, \cdots, q_{12}^{(1)}\}$ = {76.04, 33.29, 12.2, 21.79, 55.97, 44.84, 34.26, 34.29, 22.01, 13.70, 22.17, 9.71}.

5.2 Optimization using Genetic Algorithm

This phase of study deals with the problems formulated in Section 4.1. The subroutines corresponding to the formulated problems are written in C+ language and executed accordingly. The population size is taken as 60 and the number of generations as 60, so that finally a total of 3600 solutions are generated at a time using NSGA-II. The probability for crossover and mutation is taken as 0.9 and 0.1 respectively.

The problem includes 2 objectives, 14 constraints and 12 decision variables (item quantity), for 3 groups in the stores. The plots for all the 3600 solutions and the set of best solutions (Pareto-optimal front) are given below.

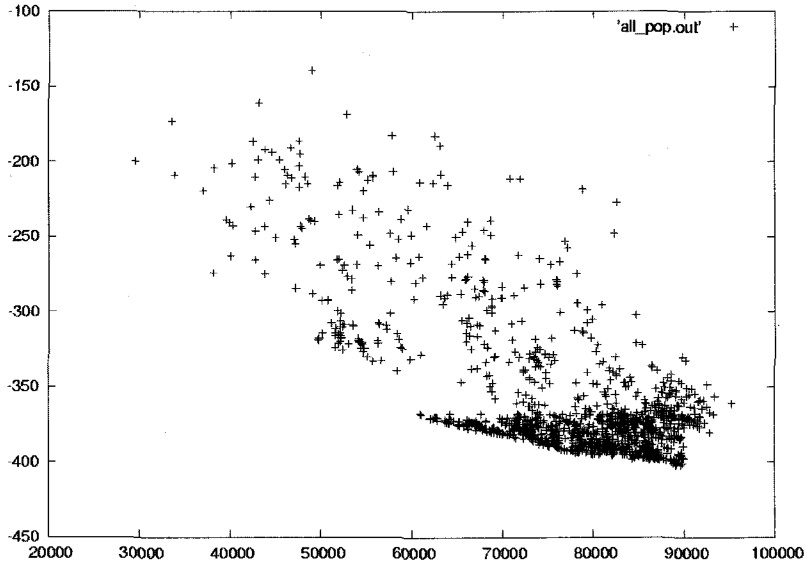

Figure 1. Solution space for all the populations

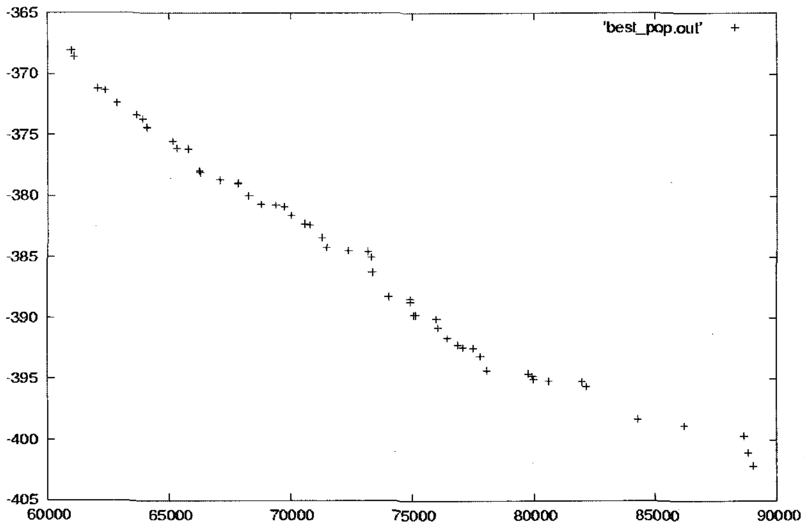

Figure 2. Pareto−optimal front of best solutions

It is observed that out of all possible 3600 solutions generated, a set of 58 solutions is obtained as Pareto-optimal solutions which are sorted in ascending order with respect to optimal values of the objective functions (quantity and cost) and tabulated in Table 3.

Table 3. Solutions of genetic algorithm approach using NSGA-II

Best Pop	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_{11}	q_{12}	Cost	Qty
1	75.17	32.22	17.30	21.89	56.22	49.25	39.02	34.04	10.32	0.77	24.84	7.05	60982.58	368.08
2	75.25	32.34	17.30	21.89	56.22	49.25	39.02	34.27	10.32	0.77	24.87	7.04	61102.67	368.55
3	75.25	32.22	17.18	21.92	56.25	49.33	38.90	38.88	7.86	1.29	25.08	7.01	62065.62	371.16
4	75.24	32.22	17.22	21.92	56.25	49.28	38.90	38.80	7.92	1.51	25.06	7.01	62373.35	371.32
5	75.25	32.35	17.30	21.88	56.03	49.25	38.90	38.31	10.32	0.75	24.87	7.17	62871.94	372.39
6	75.25	32.22	17.30	21.89	56.22	49.26	38.90	38.94	10.32	1.13	24.87	7.04	63682.19	373.34
7	75.01	32.39	17.28	21.88	56.11	49.67	38.90	38.90	10.32	1.27	24.87	7.04	63922.32	373.66
8	75.25	32.21	17.22	21.94	56.13	49.58	38.89	36.85	12.43	0.84	25.97	7.01	64094.93	374.33
9	75.25	32.21	17.22	21.94	56.12	49.58	38.90	36.85	12.43	0.84	25.93	7.10	64106.27	374.38
10	75.25	32.21	17.22	21.94	56.12	49.58	38.90	36.85	12.43	0.84	25.93	7.10	64106.27	374.38
11	75.24	32.32	17.18	21.99	56.11	49.68	38.74	36.61	14.19	0.77	26.00	6.68	65146.16	375.52
12	75.25	32.21	17.41	21.94	56.13	49.58	38.89	36.74	14.12	0.84	25.97	7.01	65325.01	376.10
13	75.24	32.46	16.33	21.94	55.77	49.01	38.91	38.84	13.78	0.85	25.99	7.01	65793.28	376.13
14	75.24	32.46	16.33	21.94	55.77	49.01	38.91	38.84	13.78	0.85	25.99	7.01	65793.28	376.13
15	75.24	32.21	17.22	21.94	55.99	49.58	38.90	38.85	14.11	0.84	25.96	7.03	66237.12	377.90
16	75.25	32.21	17.22	21.94	56.24	49.58	38.90	38.85	14.11	0.84	25.93	6.99	66264.91	378.07
17	75.25	31.71	17.41	21.94	56.13	49.58	38.89	39.32	14.12	1.30	25.97	7.05	67097.35	378.67
18	75.00	32.21	17.24	21.95	56.08	49.07	38.99	38.88	16.97	0.60	24.88	7.01	67827.07	378.88
19	75.00	32.21	17.24	21.95	56.08	49.07	38.99	38.88	16.97	0.60	24.88	7.07	67838.19	378.95
20	75.21	32.18	17.48	21.92	56.08	49.62	38.79	39.52	15.91	1.23	26.00	6.02	68271.22	379.96
21	75.18	32.18	17.48	21.91	56.08	49.62	38.79	39.50	16.60	1.23	26.00	6.08	68791.70	380.65
22	75.23	32.21	17.39	21.90	56.14	49.30	38.90	38.75	17.68	1.33	24.87	6.99	69390.77	380.71
23	75.25	32.18	17.43	21.89	56.22	49.27	38.81	38.95	15.26	2.72	25.84	7.05	69729.49	380.87
24	75.27	32.18	17.43	21.84	56.08	49.57	38.83	39.50	15.26	2.72	25.84	7.05	70014.63	381.56
25	75.19	32.22	17.24	21.94	55.80	49.66	38.93	39.49	16.17	2.65	25.96	7.01	70583.35	382.26
26	75.19	32.21	17.24	21.88	55.80	49.66	38.91	39.49	16.35	2.72	25.91	7.01	70796.10	382.37
27	75.25	32.21	17.24	21.94	56.08	49.66	38.99	38.88	17.64	2.52	25.97	7.01	71276.24	383.38
28	75.25	32.21	17.24	21.94	56.08	49.66	38.99	38.88	17.64	2.52	25.97	7.01	71276.24	383.38
29	75.25	32.21	17.24	21.94	56.08	49.66	38.99	38.88	17.64	2.52	25.98	7.01	71277.79	383.39
30	75.21	32.18	17.48	21.83	56.08	49.62	38.79	39.52	20.23	1.23	26.00	6.02	71502.09	384.18
31	75.24	32.33	17.28	21.99	56.02	49.36	38.73	35.90	24.51	0.73	26.03	6.32	72386.53	384.45
32	75.01	32.22	17.23	21.71	56.08	49.04	38.68	36.26	25.75	0.60	24.87	7.05	73186.62	384.49
33	75.01	32.22	17.23	21.73	56.08	49.04	38.68	36.94	25.51	0.60	24.87	7.05	73322.44	384.98
34	75.21	32.21	17.24	21.92	56.24	48.86	38.67	38.90	23.82	0.92	25.93	6.33	73379.26	386.25

35	75.24	32.33	17.34	21.99	56.02	49.36	38.95	39.56	24.51	0.73	25.90	6.32	74053.25	388.26
36	75.25	32.33	17.27	21.79	56.11	49.39	38.75	39.50	24.77	1.29	26.03	6.01	74936.72	388.47
37	75.25	32.18	17.39	21.79	56.08	49.62	38.88	39.50	24.77	1.26	25.97	6.01	74938.99	388.69
38	75.27	32.18	17.39	21.84	56.08	49.62	38.83	39.50	24.73	1.24	26.03	7.05	75077.91	389.77
39	75.25	32.18	17.39	21.79	56.08	49.62	38.88	39.50	24.79	1.26	26.00	7.04	75144.84	389.78
40	75.24	32.33	17.18	21.99	56.11	49.68	38.74	36.14	28.88	0.77	26.00	7.01	76013.28	390.08
41	75.27	32.18	17.48	21.84	56.08	49.60	38.79	38.69	26.61	1.22	26.00	7.05	76080.02	390.80
42	75.27	32.18	17.48	21.84	56.08	49.62	38.79	39.50	26.61	1.23	26.00	7.05	76471.22	391.65
43	75.27	32.18	17.48	21.84	56.08	49.62	38.79	39.50	27.15	1.23	26.00	7.05	76878.66	392.19
44	75.27	32.18	17.47	21.84	56.08	49.62	38.79	39.49	27.32	1.29	26.00	7.05	77088.50	392.40
45	75.23	32.18	17.26	21.94	56.08	49.66	38.77	38.76	28.32	1.32	25.97	6.99	77515.93	392.49
46	75.27	32.18	17.26	21.94	56.08	49.66	38.79	39.00	28.71	1.23	25.97	7.05	77815.38	393.15
47	75.25	32.21	17.39	21.90	56.13	49.64	38.89	39.43	29.79	0.69	25.98	6.98	78077.03	394.29
48	75.20	32.22	17.42	21.85	55.80	49.66	38.90	39.48	28.30	2.72	25.99	7.01	79784.87	394.56
49	75.25	32.21	17.48	21.94	56.23	49.66	38.54	38.88	28.88	2.72	26.00	7.01	79966.29	394.79
50	75.25	32.21	17.48	21.94	56.12	49.66	38.89	38.88	28.88	2.72	26.00	7.01	79996.06	395.02
51	75.25	32.21	17.18	21.94	56.11	49.68	38.89	38.88	28.88	3.17	26.00	7.01	80608.74	395.19
52	74.94	32.22	17.17	21.94	56.13	49.41	38.83	38.28	29.05	4.30	25.97	7.01	81982.97	395.25
53	74.94	32.22	17.17	21.94	56.13	49.41	38.83	38.59	29.10	4.30	25.97	7.01	82156.46	395.61
54	75.25	32.21	17.34	21.92	56.13	49.64	38.87	38.84	29.81	5.31	25.97	6.98	84288.83	398.27
55	75.27	32.34	17.48	21.83	56.08	49.62	38.67	39.50	27.15	7.88	26.00	7.01	86176.62	398.83
56	75.16	32.18	16.20	21.94	56.24	49.36	38.79	39.00	28.71	9.06	25.98	7.05	88644.98	399.65
57	75.25	32.18	17.35	21.92	56.13	48.65	38.90	39.51	29.79	8.49	25.90	6.98	88832.79	401.06
58	75.25	32.21	17.35	21.92	56.13	49.64	38.89	39.51	29.79	8.49	25.96	6.98	89033.76	402.13

5.3 Discussions

From the solutions of NLGP obtained in Section 5.1, the best compromise solution to the problem is found under the priority structure in which the financial goal and the item restriction goal have the highest priority ($P_1 = WS_1^3$, which involves deviational variables corresponding to item and financial restrictions). However, the calculation shows that all but the cost goals for variable q_2 and q_{12} are achieved in the *best compromise solution,* and the respective variables eventually belong to lowest priority level in this case. Since all the goals of higher priority levels are achieved, problem arises from the *compromise solution* to the decision maker. If any one or more of the goals of the assumed final priority of the *compromise solution* is not achieved, or the financial goal or the item restriction is not satisfied, then the decision maker may choose the *next best compromise solution*. The *ideal solution* can never be achieved in practical situation but it always provides a benchmark to find out compromise solutions in this method.

The most unnoticed solution for this problem by the authors of [9] was the *next best compromise solution*, where, goals for item restriction and financial restrictions were met. The sum of total quantity to be stored was found little higher (= 380.27-378.02), which is welcome along with reduction in total cost from 0.8699526 to 0.8628499 million dollar. Further, the goal for the items I-0101, I-0104, I-0202 and I-0301 are found in the first priority level (($P_1P_2P_3 = WS^1{}_1WS^2{}_2WS^3{}_3$), which the decision maker wants as a special attention to these four items. Secondly, from the point of view of preventing stock-out condition, this solution is definitely better than the best compromise solution since amount of items is little bit on the higher side satisfying the constraint. However, the calculation for cost goal reveals that the solution for variables q_1, q_2 and q_{12} fail to meet respective targets. Also, the variable q_1 belonging to highest priority level in this particular solution may not be judged as a better one by the decision maker only because of lower inventory cost.

From Table 3, using NSGA-II algorithm for GA, we obtain the pareto-optimal front comprising of 58 solutions, and when compared with the solutions achieved using GP method, we observe the following:

- Soln. No. 54 (qty = 398.27, cost = 0.8428883 million dollar) in GA is even better than *Ideal Soln.* (qty = 374.14, cost = 0.8468156 million dollar) in GP in terms of more quantity and less cost. It is true upto Soln. No. 8 in the upward direction of Table-3.
- Soln. No. 55 (qty = 398.83, cost = 0.8617662 million dollar) in GA is better than *Best Compromise Soln.* (qty = 378.02, cost = 0.8699526 million dollar) in GP in terms of more quantity and less cost. It is true upto Soln. No. 16 in the upward direction of Table 3.
- Soln. No. 55 (qty = 398.83, cost = 0.8617662 million dollar) in GA is better than *Next Best Compromise Soln.* (qty = 380.27, cost = 0.8628499 million dollar) in GP in terms of more quantity and less cost. It is true upto Soln. No. 16 in the upward direction of Table 3.

This implies that Soln No. 16 to Soln. No. 54, as found from GA, are all uniformly better than either *Ideal Solution* or two successive *Compromise Solutions* of GP or both in terms of quantity and cost.

The main advantage in using GA over GP is that the GA considers maximizing the inventory to avoid the stock-out condition to the extent possible without sacrificing the cost of inventory. The two conflicting objective functions ensure this situation by developing the Pareto-optimal front. The stock-out situation is given sufficient importance while solving through GA since the quantity to be stored has been maximised. Moreover, the decision maker can have now plenty of choices in between the Soln No. 16 to Soln. No. 54, to decide the complex decision-making process through selection of goal structure and the priority levels. The cost component in GA exceeds over GP solution in only the last three solutions (No. 56-58) (see Table 3). The maximum possible inventory that can be stored, as found in Soln. No. 58 of GA, is 402.13, but at a higher cost of 0.8903376 million dollar. The following table (Table 4) gives a comparative picture about the extreme and compromise solutions observed through GP and GA.

Table 4. Comparative Results

		GP Solutions						GA Solutions	
Item No.	Cost/Unit	Ideal Son. (q)	Cost	Best compromise Solution (q)	Cost	Next best compromise solution (q)	Cost	Max q Solution	Cost
1	55	75.19	4135.45	75.19	4135.45	76.04	4182.20	75.25	4138.72
2	65	33.08	2150.20	33.08	2150.20	33.29	2163.85	32.21	2093.88
3	75	12.20	915.00	12.88	966.00	12.20	915.00	17.35	1301.14
4	46	20.66	950.36	20.66	950.36	21.79	1002.34	21.92	1008.33
5	160	54.00	8640.00	54.00	8640.00	55.97	8955.20	56.13	8980.60
6	195	43.81	8542.95	43.81	8542.95	44.84	8743.80	49.64	9680.77
7	140	34.11	4775.40	34.11	4775.40	34.26	4796.40	38.89	5444.15
8	450	34.13	15358.50	34.13	15358.50	34.29	15430.50	39.51	17778.23
9	750	22.01	16507.50	23.66	17745.00	22.01	16507.50	29.79	22344.12
10	1400	13.07	18298.00	13.72	19208.00	13.70	19180.00	8.49	11891.33
11	120	22.17	2660.40	22.95	2754.00	22.17	2660.40	25.96	3115.40
12	180	9.71	1747.80	9.83	1769.40	9.71	1747.80	6.98	1257.09
Total	–	374.14	84681.56	378.02	86995.26	380.27	86284.99	402.13	89033.76

6. Concluding Remarks

The total solutions obtained from NLGP method using IP_D concept are small in number (only six), and the priority structure suggested by the decision-maker can not be strictly maintained. However, the *best compromise solution* achieves all except two cost goals with two lowest priority variables, and hence may be accepted by the decision maker. The *next best compromise solution*, although having lower inventory cost than the *best compromise solution*, violates the cost goal for one highest priority footwear sub-category, namely, I-0101.

On the other hand, the solutions obtained from GA are large in number so that there are plenty of options to be chosen from. Although the priority structure is not considered in this case, all the goals used in NLGP are incorporated in the formulation and thus, almost all the solutions obtained from GA are better than *best compromise solution*. Therefore, the decision maker may make his choice from plenty of alternatives based on his own priority considerations. To conclude, the highlighting points arising out of this study are summarized below.

- The GP method has failed to maintain the imposed priority structure for different goals as desired by the decision-maker. However, the goals have been achieved.
- The GP method has failed to identify the *best compromise solution* in terms of cost minimization. For cost minimization problems, the IP_D method with respect to ideal solution cannot work with quantity (decision variables) only, unless cost components are included in the formulae.
- The GA generated large number of solutions (populations) at the Pareto-optimal front, each of which may be acceptable by the decision maker depending on the priority structure.
- The GA takes care of stock-out prevention by maximizing the inventory without sacrificing the cost.
- The GA is able to find out the solutions that are uniformly better than the solutions proposed by GP method.

Reference

[1] Alptekinoglu, A. and C. S., Tang, "A model for analyzing multi-channel distribution systems," *European Journal of Operational Research* 163 (2005), 802-824.

[2] Basu M., D. Ghosh, and K. Banerjee, "A Solution Procedure For Solving Multi-item Inventory Problem," *International Journal of Management and Systems* 15, 1 (1997), 53-68.

[3] Charnes, A. and W. W. Cooper, *Management Models and Industrial Application of Linear Programming,* John Wiley & Sons. Inc. New York (1961).

[4] Charnes, A. and B. Collomb, "Optimal economic stabilization policy: linear goal-interval programming models," *Socio-Economic Planning Sciences* 6, 4 (1972), 431-435.

[5] Charnes A., W. W. Cooper, J. Harrald, K. Karwan, and W. Wallace, "A goal interval programming model for resource allocation in marine environment protection problem," *J. of Envir. Econ. Mgmt.* 3 (1976), 347-362.

[6] Deb, K., *Multi-objective Optimization using Evolutionary Algorithms,* John Wiley & Sons Ltd. New York (2001).

[7] Mitchell, M., *An Introduction to Genetic Algorithms,* The MIT Press, Cambridge, MA, USA (1998).

[8] Mondal S. and M. Maiti, "Multi-item fuzzy EOQ models using genetic algorithm," *Computers and Industrial Engineering* 44 (2002), 105-117.

[9] Panda, S., S. Senapati, K. Banerjee, and M. Basu, "Determination of EOQ of multi-item inventory problems through nonlinear goal programming," *Advanced Modeling and Optimization* 7, 2 (2005), 169-176.