

## Solving an aggregate production planning problem by using multi-objective genetic algorithm (MOGA) approach

Ripon Kumar Chakraborty<sup>a\*</sup> and Md. A. Akhtar Hasin<sup>b</sup>

<sup>a</sup>Department of Industrial & Production Engineering, Rajshahi University of Engineering & Technology, Rajshahi-6204, Bangladesh

<sup>b</sup>Department of Industrial and Production Engineering, Bangladesh University of Science and Technology (BUET), Dhaka-1000, Bangladesh

### ARTICLE INFO

#### Article history:

Received 5 September 2012

Received in revised format

25 September 2012

Accepted September 27 2012

Available online

27 September 2012

#### Keywords:

Multi-objective optimization

Genetic algorithm

Aggregate production planning

### ABSTRACT

In hierarchical production planning system, Aggregate Production Planning (APP) falls between the broad decisions of long-range planning and the highly specific and detailed short-range planning decisions. This study develops an interactive Multi-Objective Genetic Algorithm (MOGA) approach for solving the multi-product, multi-period aggregate production planning (APP) with forecasted demand, related operating costs, and capacity. The proposed approach attempts to minimize total costs with reference to inventory levels, labor levels, overtime, subcontracting and backordering levels, and labor, machine and warehouse capacity. Here several genetic algorithm parameters are considered for solving NP-hard problem (APP problem) and their relative comparisons are focused to choose the most auspicious combination for solving multiple objective problems. An industrial case demonstrates the feasibility of applying the proposed approach to real APP decision problems. Consequently, the proposed MOGA approach yields an efficient APP compromise solution for large-scale problems.

© 2012 Growing Science Ltd. All rights reserved

## 1. Introduction

Aggregate production planning is associated with the determination of inventory, production and work force levels to consider fluctuating demand needs over a planning horizon, which ranges from six months up to a year. Typically, the planning horizon includes the next seasonal peak in demand. The planning horizon can be divided into periods. For instance, a one-year planning horizon could consist of six one-month periods plus two three-month periods. We may consider a fixed value for the physical resources of the firm during the planning horizon of interest and the planning attempt is oriented towards the best utilization of those resources, given the external demand needs. Since it is usually impractical to consider every fine detail related to the production process while maintaining such a long planning horizon, it is obligatory to aggregate the information being processed. The aggregate production approach is forecasted on the existence of an aggregate unit of production, such as the "average" item, or in terms of weight, volume, production time, or dollar value. Plans are based on aggregate demand for one or more aggregate items. Once the aggregate production plan is created,

\* Corresponding author. +88-01911769364

E-mail: ripon\_ipebuet@yahoo.com (R. Kumar Chakraborty)

constraints are applied on the detailed production scheduling process, which decides the specific quantities to be produced of each individual item.

APP has attracted considerable interest from both practitioners and academics (Shi & Haase, 1996). For solving APP problems, certain constraints are imposed which demand constraint optimization. Ioannis (2009) described a novel genetic algorithm for the problem of constrained optimization. His model was a modified version of the genetic operators namely crossover and mutation. These new version preserve the feasibility of the trial solutions of the constrained problem that are encoded in the chromosomes. Bunnag and Sun (2005) presented a stochastic optimization method, referred to as a Genetic Algorithm (GA), for solving constrained optimization problems over a compact search domain. It was a real-coded GA, which converges in probability to the optimal solution. The constraints were treated through a repair operator. A specific repair operator was included for linear inequality constraints. Summanwar et al. (2002) introduced a method for constrained optimization using a modified multi-objective algorithm. Their algorithm treats the constraints as objective functions and handles them using the concept of Pareto dominance. The population members were ranked by two different methods: first ranking is based on objective function value and the second ranking is based on Pareto dominance of the population members.

When we solve APP problem, we have to face with uncertain market demands and capacities in production environment, imprecise process times, and other factors introducing inherent uncertainty to the solution. Using deterministic and stochastic frameworks in such conditions may not lead to desirable results (Aliev et al., 2007). Aliev et al. (2007) developed a fuzzy integrated multi-period and multi-product production and distribution model in supply chain where the model was modeled in terms of fuzzy programming and the solution was provided by genetic optimization.

Genetic Algorithm (GA) normally provides a series of alternative solutions for various GA parameter values. The decision-maker can find alternative optimal solutions from a series of alternative values (Sharma & Jana, 2009). In order for GAs to surpass their more traditional cousins in the quest for robustness, they must vary in some very fundamental ways (Goldberg, 1989). Four differences separate GAs from more traditional optimization techniques and those are, direct manipulation of a coding, searching from a population rather than a single point, following a blind searching technique and finally search using stochastic operators, not deterministic rules. It can be quite efficient to combine GA with other optimization methods. GA seems to be quite good for finding generally good global solutions, but quite inefficient at locating the last few mutations to determine the absolute optimum. Other techniques (such as simple hill climbing) are quite efficient at finding absolute optimum in a limited region. Alternating GA and hill climbing can improve the efficiency of GA while overcoming the lack of robustness of hill climbing. For solving Multiple Objective problems GA could generate the most optimum value (Yeh & Chuang, 2011).

Multi-objective optimization, multi-objective programming or Pareto optimization also known as multi-criteria or multi-attribute optimization is the process of simultaneously optimizing two or more conflicting objectives subject to certain constraints (Cai & Wang, 2006). These problems have absorbed many researchers using traditional techniques of optimization and search as well as GAs (Schaffer, 1985). On the other hand, Lai and Hwang (1992) developed an auxiliary multiple objective linear programming (MOLP) model for solving a PLP problem with imprecise objective and/or constraint coefficients. Yeh and Chuang (2011) used multi-objective GA for partner selection in green supply chain problems. In their work, they involved four objectives such as cost, time, product quality and green appraisal score for optimization or minimization. In order to solve these conflicting objectives, they adopted two multi-objective genetic algorithms to find the set of Pareto-optimal solutions, which utilized the weighted sum approach, which could generate more number of solutions. This implies Pareto optimality is more suitable for multi-objective optimization cases. Number of Pareto-optimal solutions is also a determinant for suitability justification (Yeah & Chuang, 2011).

Again, with the consideration of NP-hard problems Moghaddam & Safaei (2006) presented a genetic algorithm (GA) for solving a generalized model of single-item resource constrained aggregate production planning (APP) with linear cost functions. In their paper, they developed a new genetic algorithm with effective operators and integer representation. Most recently, Ramezani et al. (2012) concentrated on multi-period, multi-product and multi-machine systems with setup decisions. In their study, they developed a mixed integer linear programming (MILP) model for general two-phase aggregate production planning systems. Due to NP-hard class of APP, they implemented a genetic algorithm and Tabu search for solving this problem.

Throughout the review, it is obvious that there have been a long evolution phase for GA algorithms. Yet the researchers obstinately keep on this and they got newer dimension. Here the authors become optimistic enough after reviewing all the literatures since there are good opportunities for future contributions. Here, the authors considered multiple objectives for multi period and multi product APP problem. However, the distinction lies in the followed approach. We used five scenarios simultaneously with different GA options for solving multiple objectives. A detailed comparison is also placed to choose the perfect combination of GA parameters. In the previous works with GA for APP, there not any single application of escalating factors for any little uncertainty or imprecise costs. This work develops a novel interactive MOGA approach considering escalating factors as well. The proposed approach attempts to minimize total costs in terms of inventory levels, labor levels, overtime, subcontracting and backordering levels, and labor, machine and warehouse capacity.

The rest of this paper is organized as follows: Section 2 describes the problem, details the assumptions, and formulates the problem. Section 2 also focused on the multiple objectives for the APP case and considered parameters for solving this MOGA approach. Subsequently, Section 3 presents an industrial case designed on Bangladeshi perspective to implement the feasibility of applying the proposed Multiple Objective Genetic Algorithm (MOGA) approach to real APP decision problems. Next, Section 4 discusses the results and findings for the practical application of the proposed PLP approach. Conclusions are finally drawn in Section 6.

## 2. Problem formulation

### 2.1 Problem description & notation

The multi-product APP problem can be described as follows. Assume that a company manufactures  $N$  kinds of products to meet market demand over a planning horizon  $T$ . This APP problem focuses on developing an interactive MOGA approach to determine the optimum aggregate plan for meeting forecasted demand by adjusting regular and overtime production rates, inventory levels, labor levels, subcontracting and backordering rates, and other controllable variables. Based on the above characteristics of the considered APP problem, the mathematical model herein is developed on the following assumptions.

- i. The values of all parameters are certain over the next  $T$  planning horizon.
- ii. The escalating factors in each of the costs categories are certain over the next  $T$  planning horizon.
- iii. Actual labor levels, machine capacity and warehouse space in each period cannot exceed their respective maximum levels.
- iv. The forecasted demand over a particular period can be either satisfied or backordered, but the backorder must be fulfilled in the next period.

The following notation is used after reviewing the literature and considering practical situations (Wang & Liang, 2004; Masud & Hwang, 1980; Wang & Fang, 2001).

$D_{nt}$	Forecasted demand for $n^{th}$ product in period $t$ (units)
$a_{nt}$	Regular time production cost per unit for $n^{th}$ product in period $t$ (Tk./unit)
$Q_{nt}$	Regular time production for $n^{th}$ product in period $t$ (units)
$i_a$	Escalating factor for regular time production cost (%)
$b_{nt}$	Overtime production cost per unit for $n^{th}$ product in period $t$ (Tk./unit)
$O_{nt}$	Overtime production for $n^{th}$ product in period $t$ (units)
$i_b$	Escalating factor for overtime production cost (%)
$c_{nt}$	Subcontracting cost per unit of $n^{th}$ product in period $t$ (Tk./unit)
$S_{nt}$	Subcontracting volume for $n^{th}$ product in period $t$ (units)
$i_c$	Escalating factor for subcontract cost (%)
$d_{nt}$	Inventory carrying cost per unit of $n^{th}$ product in period $t$ (Tk./unit)
$I_{nt}$	Inventory level in period $t$ for $n^{th}$ product (units)
$i_d$	Escalating factor for inventory carrying cost (%)
$e_{nt}$	Backorder cost per unit of $n^{th}$ product in period $t$ (Tk./unit)
$B_{nt}$	Backorder level for $n^{th}$ product in period $t$ (unit)
$i_e$	Escalating factor for backorder cost (%)
$K_t$	Cost to hire one worker in period $t$ (Tk./man-hour)
$H_t$	Worker hired in period $t$ (man-hour)
$m_t$	Cost to layoff one worker in period $t$ (Tk./man-hour)
$F_t$	Workers laid off in period $t$ (man-hour)
$i_f$	Escalating factor for hire and layoff cost (%)
$i_{nt}$	Hours of labor per unit of $n^{th}$ product in period $t$ (man-hour/unit)
$r_{nt}$	Hours of machine usage per unit of $n^{th}$ product in period $t$ (machine-hour/unit)
$V_{nt}$	Warehouse spaces per unit of $n^{th}$ product in period $t$ (ft <sup>2</sup> /unit)
$W_{tmax}$	Maximum labor level available in period $t$ (man-hour)
$M_{tmax}$	Maximum machine capacity available in period $t$ (machine-hour)
$V_{tmax}$	Maximum warehouse space available in period $t$ (ft <sup>2</sup> )

## 2.2 Multi-Objective Genetic Algorithm (MOGA) Model

### 2.2.1 Multi-Objective functions

Most practical decisions made to solve APP problems usually consider total costs. The proposed MOGA targeted three objective functions. First, it selected total costs as objective function, after reviewing the literature and considering practical situations (Masud & Hwang, 1980; Saad, 1982; Wang & Fang, 2001). The total costs are the sum of the production costs and the costs of changes in labor levels over the planning horizon  $T$ . Accordingly, the objective function of the proposed model is as follows:

$$\text{Min } Z_1 = \sum_{n=1}^N \sum_{t=1}^T [a_{nt} Q_{nt} (1 + i_a)^t + b_{nt} O_{nt} (1 + i_b)^t + c_{nt} S_{nt} (1 + i_c)^t + d_{nt} I_{nt} (1 + i_d)^t + e_{nt} B_{nt} (1 + i_e)^t] + \sum_{t=1}^T (K_t H_t + m_t F_t) (1 + i_f)^t$$

Here the first five terms are used to calculate production costs. The production costs include five components-regular time production, overtime, and subcontracts, carrying inventory and backordering cost. The later portion specifies the costs of change in labor levels, including the costs of hiring and lay off workers. Escalating factors were also included for each of the cost categories. Again, for minimizing carrying and backordering cost ( $Z_2$ ) and minimizing rate of change in labor levels ( $Z_3$ ) the following objective functions are considered.

$$\text{Min } Z_2 = \sum_{n=1}^N \sum_{t=1}^T [d_{nt} I_{nt} (1 + i_d)^t + e_{nt} B_{nt} (1 + i_e)^t] \text{ and } \text{Min } Z_3 = \sum_{t=1}^T (H_t - F_t)$$

### 2.2.2 Constraints

Constraints on carrying inventory:

$$I_{nt} - B_{nt} = I_{n(t-1)} - B_{n(t-1)} + Q_{nt} + O_{nt} + S_{nt} - D_{nt} \text{ for } \forall n, \forall t \quad (1)$$

$$I_{nt} \geq I_{ntmin} \text{ for } \forall n, \forall t \quad (2)$$

$$B_{nt} \leq B_{ntmax} \text{ for } \forall n, \forall t \quad (3)$$

where,  $D_{nt}$  denotes the imprecise forecast demand of the  $n^{th}$  product in period  $t$ . In real-world APP decision problems, the forecast demand  $D_{nt}$  cannot be obtained precisely in a dynamic market. The sum of regular and overtime production, inventory levels, and subcontracting and backorder levels essentially should equal the market demand, as in first constraint Equation. Demand over a particular period can be either met or backordered, but a backorder must be fulfilled in the subsequent period.

Constraints on Labor levels:

$$\sum_{n=1}^N i_{n(t-1)} (Q_{n(t-1)} + O_{n(t-1)}) + H_t - F_t = \sum_{n=1}^N i_{nt} (Q_{nt} + O_{nt}) \text{ for } \forall t \quad (4)$$

$$\sum_{n=1}^N i_{nt} (Q_{nt} + O_{nt}) \leq W_{tmax} \text{ for } \forall t \quad (5)$$

Here in the fourth constraint, equation represents a set of constraints in which the labor levels in period  $t$  equal the labor levels in period  $t-1$  plus new hires less layoffs in period  $t$ . Actual labor levels cannot exceed the maximum available labor levels in each period, as in fifth equation. Maximum available labor levels are imprecise, owing to uncertain labor market demand and supply.

Constraints on Machine capacity & Warehouse space:

$$S_{nt} \leq S_{ntmax} \text{ for } \forall n, \forall t \quad (6)$$

$$\sum_{n=1}^N r_{nt} (O_{nt} + Q_{nt}) \leq M_{tmax} \text{ for } \forall t \quad (7)$$

$$\sum_{n=1}^N V_{nt} I_{nt} \leq V_{tmax} \text{ for } \forall t \quad (8)$$

Eq. (6-8) represent the limits of actual machine and warehouse capacity in each period. Non-negativity Constraints on decision variables are:

$$Q_{nt}, O_{nt}, S_{nt}, I_{nt}, B_{nt}, H_t, F_t \geq 0 \text{ for } \forall n, \forall t \quad (9)$$

### 2.3 Outline of the Basic MOGA Model

Step 1: Generate random population of  $n$  chromosomes (suitable solutions for the problem)

Step 2: Evaluate simultaneously the Multiple fitness  $f(x)$  of each chromosome  $x$  in the population

Step 3: Create a new population by repeating four steps (Selection, Crossover, Mutation and Acceptation) until the new population is complete.

Step 4: Use new generated population for a further run of algorithm

Step 5: If the stopping condition is satisfied, stop, and return the best solution in current population

Step 6: If the stopping condition is not satisfied then go to step 2 & follow loop.

## 2.4 Multi-Objective Genetic Algorithm (MOGA) Parameters

### 2.4.1 Crossover Options

Crossover options specify how the GA combines two individuals, or parents, to form a crossover child for the next generation. Here the authors choose five different crossover options for five scenarios.

- I. **Scattered crossover** creates a random binary vector and selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child. For example, if p1 and p2 are the parents such as p1 = [a b c d e f g h] and p2 = [1 2 3 4 5 6 7 8] and the binary vector is [1 1 0 0 1 0 0 0], then the function returns the following child1 = [a b 3 4 e 6 7 8]
- II. **Single point crossover** chooses a random integer  $n$  between 1 and number of variables and then selects vector entries numbered less than or equal to  $n$  from the first parent and selects vector entries numbered greater than  $n$  from the second parent. For example, if p1 and p2 are the parents such as p1 = [a b c d e f g h] and p2 = [1 2 3 4 5 6 7 8] and the crossover point is 3, the function returns the following child = [a b c 4 5 6 7 8]
- III. **Two points crossover** selects two random integer  $m$  and  $n$  between 1 and number of variables. The function selects Vector entries numbered less than or equal to  $m$  from the first parent, vector entries numbered from  $m+1$  to  $n$ , inclusive, from the second parent, vector entries numbered greater than  $n$  from the first parent. The algorithm then concatenates these genes to form a single gene. For example, if p1 and p2 are the parents such as p1 = [a b c d e f g h] and p2 = [1 2 3 4 5 6 7 8] and the crossover points are 3 and 6, the function returns the following child = [a b c 4 5 6 g h]
- IV. **Arithmetic crossover** is a crossover operator that linearly combines two parent chromosome vectors to produce two new offspring according to the following equations:  

$$\text{Offspring1} = a * \text{Parent1} + (1-a) * \text{Parent2}$$

$$\text{Offspring2} = (1-a) * \text{Parent1} + a * \text{Parent2}$$
 where  $a$  is a random weighting factor (chosen before each crossover operation).
- V. **Heuristic Crossover** is a crossover operator that uses the fitness values of the two parent chromosomes to determine the direction of the search. The offspring are created according to the following equations where  $r$  is a random number between 0 and 1.  

$$\text{Offspring1} = \text{Best Parent} + r * (\text{Best Parent} - \text{Worst Parent})$$

$$\text{Offspring2} = \text{Best Parent}$$

### 2.4.2 Mutation Options

Mutation options specify how the genetic algorithm makes small random changes in the individuals in the population to create mutation children. Mutation provides genetic diversity and enables the GA to search a broader space. Here the authors use Constraint dependent mutation & Adapt feasible mutation options. Adaptive Feasible randomly generates directions that are adaptive with respect to the last

successful or unsuccessful generation. The feasible region is bounded by the constraints and inequality constraints. A step length is chosen along each direction so that linear constraints and bounds are satisfied.

### 2.4.3 Creation function

Creation function creates the initial population for GA. Here the authors choose Feasible population & Constraint dependent options. Feasible population creates a random initial population that satisfies all bounds and linear constraints. It is biased to create individuals that are on the boundaries of the constraints, and to create well-dispersed populations. This is the default if there are linear constraints.

### 2.4.4 Selection Options

Selection options specify how the genetic algorithm chooses parents for the next generation. Here the authors used only Tournament selection option for tournament size 2 & 4. Tournament selection chooses each parent by choosing Tournament size players at random and then choosing the best individual out of that set to be a parent.

### 2.4.5 Migration Options

Migration options specify how individuals move between subpopulations. Migration occurs if we set **Population size** to be a vector of length greater than 1. When migration occurs, the best individuals from one subpopulation replace the worst individuals in another subpopulation. Individuals that migrate from one subpopulation to another are copied. They are not removed from the source subpopulation.

## 3. Model implementation

### 3.1 Case description

Comfit Composite Knit Limited (CCKL) was used as a case study to demonstrate the practicality of the proposed methodology. The Comfit Composite Knit Limited is the sister concern of Youth Group, which is one of the pioneer company of Ready Made Garments (RMG) sector in Bangladesh. This company readily produced knit ware items among them some are fancy & some are expensive. The jacket items as well as cardigan items are very expensive and we need significant amount of time and cost incurring manufacturing items. Therefore, it needs a lot of precise observations & perfect manufacturing practices to catch up the market & satisfy the buyers within specified lead time. Since they are the most expensive items, major concentration was on one particular style of hooded jacket (Product 1) & another special type of ladies cardigan (Product 2).

The APP decision problem for CCKL's Knit garments manufacturing plant presented here focuses on developing an interactive GA approach for minimizing total costs. The planning horizon is 2 months long, including May and June. The model includes two types of knit ware items, namely the hooded jacket (Product 1) and special type of ladies cardigan (Product 2). According to the preliminary environmental information, Table 1 and Table 2 summarize the forecast demand, related operating cost, and capacity data used in the CCKL case. Other relevant data are as follows.

- I. Initial inventory in period 1 is 500 units of product 1 and 200 units of product 2. End inventory in period 2 is 400 units of product 1 and 300 units of product 2.
- II. Initial labor level is 225 man-hour. The costs associated with hiring and layoffs are Tk.22 and Tk.8 per worker per hour, respectively.
- III. Hours of labor per unit for any periods are fixed to 0.033 man-hour for product 1 and 0.05 man-hour for product 2. Hours of machine usage per unit for each of the two planning periods are 0.1

machine-hours for product 1 and 0.08 machine-hours for product 2. Warehouse spaces required per unit are 1 square feet for product 1 and 1.5 square feet for product 2.

IV. The expected escalating factor in each of the costs categories are 1%.

**Table 1**

Forecasted demand, maximum labor, machine, warehouse capacity, back order level, subcontracted volume & minimum Inventory data

Item (Units)	Period		Items	Period	
	1	2		1	2
$D_{1t}$ (pieces)	1400	3000	$S_{1tmax}$ (pieces)	200	350
$D_{2t}$ (pieces)	1600	800	$S_{2tmax}$ (pieces)	100	100
$W_{tmax}$ (man-hours)	225	225	$I_{1tmin}$ (pieces)	300	500
$M_{tmax}$ (machine-hours)	400	500	$I_{2tmin}$ (pieces)	150	200
$V_{tmax}$ (ft <sup>2</sup> )	1000	1000	$B_{1tmax}$ (pieces)	200	600
			$B_{2tmax}$ (pieces)	150	100

**Table 2**

Related Operating cost data for the CCKL case

Product	$a_{nt}$ (tk./unit)	$b_{nt}$ (tk./unit)	$c_{nt}$ (tk./unit)	$d_{nt}$ (tk./unit)	$e_{nt}$ (tk./unit)
1	22	40	27	3.5	42
2	20	40	30	4	47

The authors used MATLAB computer software to solve the proposed MOGA approach for the CCKL case. Total five runs were implemented considering five scenarios with different MOGA parameters shown in Table 3. Table 4 lists the multiple objective values for five MOGA runs through MATLAB. Finally, since from Table 4 it is clear that the least cost is achieved in the fourth scenario so in Table 5 lists the entire initial APP plan for the CCKL case based on the present information for that fourth scenario.

**Table 3**

Different Genetic Algorithm options used for five scenarios

MOGA Parameters/ Options	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
Population Type	Double	Double	Double	Double	Double
Population Size	50	50	360	360	360
Creation Function	Constraint Dependent	Feasible Population	Constraint Dependent	Feasible Population	Feasible Population
Mutation	Constraint Dependent	Adapt Feasible	Adapt Feasible	Constraint Dependent	Adapt Feasible
Crossover	Two point	Heuristic	Arithmetic	Scattered	Single Point
Migration (Fraction)	Forward (0.2)	Both (0.2)	Both (0.5)	Forward (0.2)	Both (0.5)
Reproduction (Fraction)	Crossover (0.8)	Crossover (0.5)	Crossover (0.5)	Crossover (0.8)	Crossover (0.8)
Selection Function (Size)	Tournament (2)	Tournament (4)	Tournament (4)	Tournament (2)	Tournament (2)
Distance Measure Function	Crowding	Crowding	Crowding	Crowding	Crowding
Pareto Front Population Fraction	0.35	0.35	0.35	0.35	0.35
Iteration needed to complete	103 Generations	260 Generations	134 Generations	111 Generations	152 Generations



**Table 4**  
Multi-objective values for different scenarios

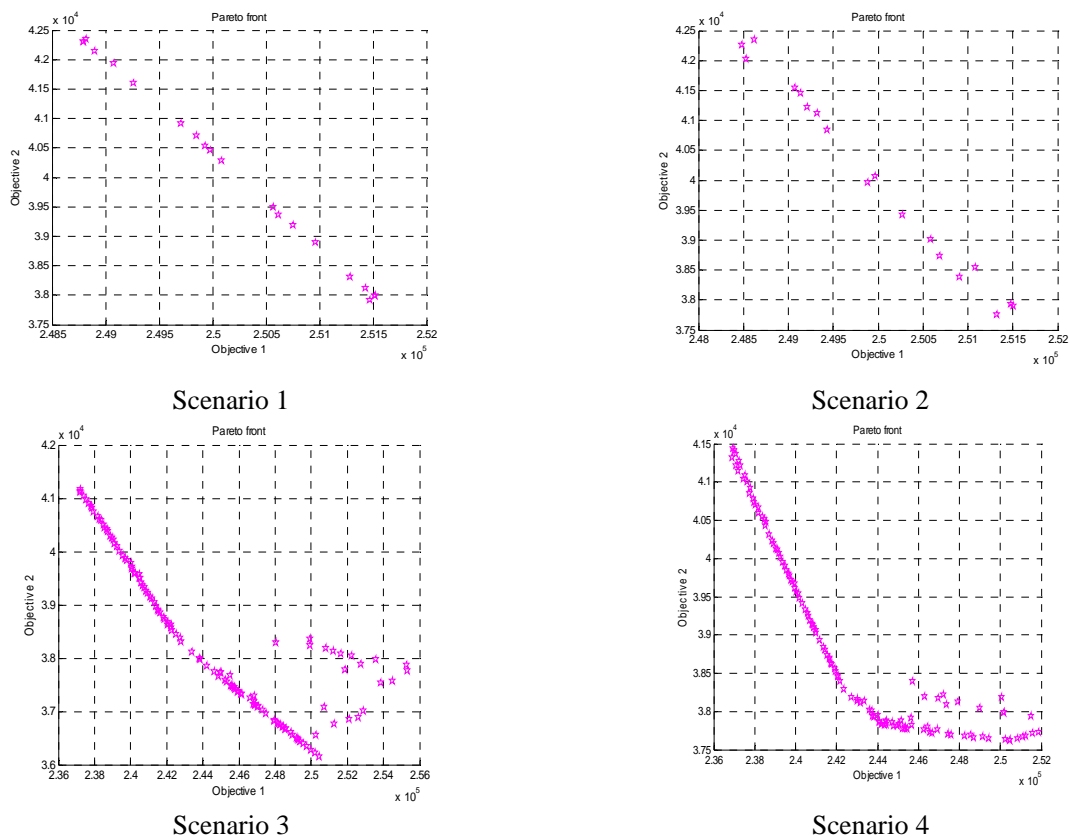
Objectives	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
$Z_1$	2,37,170.4	2,50,581	2,37,205.21	2,36,893.43	2,36,760.67
$Z_2$	41,256.88	39,017.74	41,140.35	41,319.66	41,501.20
$Z_3$	111.73	109.18	112.28	112.36	112.03
Total Cost	2,78,539	2,89,707.90	2,78,457.8	2,78,325.4	2,78,373.90

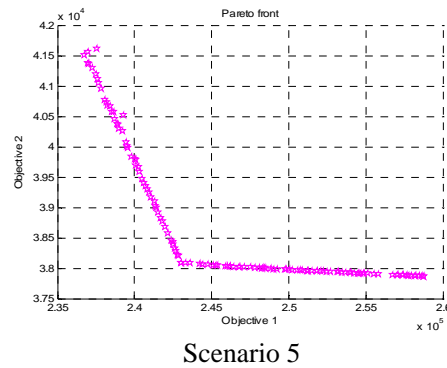
**Table 5**  
Initial multi-product & multi-period APP plan for the CCKL case (**Fourth Scenario**)

Items (Product 1)	Period		Items (Product 2)	Period	
	1	2		1	2
$Q_{1t}$ (Units)	571	1034	$Q_{2t}$ (Units)	664	446
$O_{1t}$ (Units)	571	1021	$O_{2t}$ (Units)	670	445
$S_{1t}$ (Units)	186	334	$S_{2t}$ (Units)	85	95
$I_{1t}$ (Units)	432	524	$I_{2t}$ (Units)	159	213
$B_{1t}$ (Units)	5	583	$B_{2t}$ (Units)	140	95
$H_t$ (man-hours)	104.4	170.67	$F_t$ (man-hours)	0	162.7

**4. Results and findings**

The proposed MOGA approach can solve most real-world APP problems through an interactive decision making process. The proposed model constitutes a systematic framework that facilitates the decision-making process. The proposed MOGA approach outputs more wide-ranging decision information than other models. The proposed MOGA approach focuses on the multi-periods and multi-products (product family) problems in an APP decision making process.





**Fig 1.** Generated Pareto Fronts for five Scenarios (Source: MATLAB)

From Fig 1 several characteristics of this proposed MOGA approach can be drawn. Since the concerned APP problem has multiple objectives so Pareto optimization must be considered. For scenario 1 & 2, the Pareto front is vastly dispersed & their score diversity was very poor. Again for scenario 3 & 4 it looks pretty but it also dispersed compared to scenario 5. Therefore, it may conclude that the fifth scenario is mostly optimum though it have narrow higher cost than scenario 4. The proposed approach also provides information on alternative strategies for overtime, subcontracting, inventory, backorders, and hiring and layoffs workers, in response to variations in forecast demand. Additionally, the proposed model considers the actual limitations in labor, machine, and warehouse capacity. This proposed MOGA approach also can helps to determine optimum solution even it is NP (nondeterministic polynomial) hard problems.

## 5. Conclusions

The APP decision aims to set overall production levels for each product category to meet future demand, frequently from 3 to 18 months ahead, such that APP also determines the appropriate resources to be used. This work presents a novel interactive MOGA approach for solving multi product and multi period APP decision problems with the forecast demand, related operating costs, and capacity. The proposed MOGA approach yields an efficient APP compromise solution and overall degree of DM satisfaction with determined goal values. Moreover, the proposed approach provides a systematic framework that facilitates the decision-making process, enabling a DM to interactively modify the MOGA parameters and related model parameters until a satisfactory solution is obtained. Different Genetic Algorithm options have been considered in this APP problem, which could make an impression for the future researchers to choose the suitable combination for solving multiple objective problems. Consequently, the proposed approach is expected to be suitable for making real world APP decisions.

## References

- Aliev, R.A., Fazlollahi, B., Guirimov, B.G., & Aliev, R.R. (2007). Fuzzy-genetic approach to aggregate production–distribution planning in supply chain management. *Information Sciences*, 177, 4241–4255.
- Baykasoglu, A. (2001). MOAPPS 1.0: Aggregate production planning using the multiple- objective tabu search. *International Journal of Production Research*, 39, 3685–3702.
- Bellman, R.E., & Zadeh, L.A. (1970). Decision-making in a fuzzy environment. *Management Science*, 17, 141–164.
- Buckley, J.J. (1988). Possibilistic linear programming with triangular fuzzy numbers. *Fuzzy Sets and Systems*, 26, 135–138.

- Buckley, J.J. (1989). Solving possibilistic linear programming problems. *Fuzzy Sets and Systems*, 31, 329–341.
- Bunnag, D., & Sun, M. (2005). Genetic algorithm for constrained global optimization in continuous variables. *Applied Mathematics and Computation*, 171, 604–636.
- Cai, Z., & Wang, Y. (2006). A Multi-objective Optimization-Based Evolutionary Algorithm for Constrained Optimization. *IEEE Transactions on Evolutionary Computation*, 10 (6), 658–675.
- Dobos, I. (2003). Optimal production–inventory strategies for a HMMS-type reverse logistics system. *International Journal of Production Economics*, 81–82, 351–360.
- Fung, R.Y.K., Tang, J., & Wang, D. (2003). Multiproduct aggregate production planning with fuzzy demands and fuzzy capacities. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 33 (3), 302–313.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization & Machine Learning*. Pearson Education Pvt. Ltd., Singapore.
- Gnoni, M.G., Iavagnilio, R., Mossa, G., Mummolo, G., & Leva, A.D. (2003). Production planning of a multi-site manufacturing system by hybrid modelling: A case study from the automotive industry. *International Journal of Production Economics*, 85 (2), 251–262.
- Holt, C.C., Modigliani, F., & Simon, H.A. (1955). Linear decision rule for production and employment scheduling. *Management Science*, 2, 1–30.
- Hsu, H.M., & Wang, W.P. (2001). Possibilistic programming in production planning of assemble-to-order environments. *Fuzzy Sets and Systems*, 119, 59–70.
- Hussein, M.L. (1998). Complete solutions of multiple objective transportation problems with possibilistic coefficients. *Fuzzy Sets and Systems*, 93, 293–299.
- Hwang, C.L. & Yoon, K. (1981). *Multiple Attribute Decision Making: Methods and Applications*. Springer, Berlin.
- Ioannis, G.T. (2009). Solving constrained optimization problems using a novel genetic algorithm. *Applied Mathematics and Computation*, 208, 273–283.
- Inuiguchi, M., & Sakawa, M. (1996). Possible and necessary efficiency in possibilistic multiobjective linear programming problems and possible efficiency test. *Fuzzy Sets and Systems*, 78, 231–241.
- Jensen, H.A., & Maturana, S. (2002). A possibilistics decision support system for imprecise mathematical programming problems. *International Journal of Production Economics*, 77 (2), 145–158.
- Lai, Y.J. and Hwang, C.L. (1992). A new approach to some possibilistic linear programming problems. *Fuzzy Sets and Systems*, 49, 121–133.
- Lee, Y.Y. (1990). *Fuzzy set theory approach to aggregate production planning and inventory control*. Ph.D. Dissertation, Department of I.E., Kansas State University.
- Masud, S.M., & Hwang, C.L. (1980). An aggregate production planning model and application of three multiple objective decision methods. *International Journal of Production Research*, 18, 741–752.
- Moghaddam, R.T., & Safaei, N. (2006). Solving a generalized aggregate production planning problem by genetic algorithms. *Journal of Industrial Engineering International*, 2(1), 53–64.
- Ramezani, R., Rahmani, D., & Barzinpour, F. (2012). An aggregate production planning model for two phase production systems: Solving with genetic algorithm and tabu search. *Expert Systems with Applications*, 39, 1256–1263.
- Saad, G. (1982). An overview of production planning model: Structure classification and empirical assessment. *International Journal of Production Research*, 20, 105–114.
- Schaffer, J.D. (1985). *Multiple objective optimization with vector evaluated genetic algorithms*. Proceedings of an International Conference on Genetic Algorithms and Their Applications, 93–100.
- Sharma, D.K., & Jana, R.K. (2009). Fuzzy goal programming based genetic algorithm approach to nutrient management for rice crop planning. *International Journal of Production Economics*, 121, 224–232.
- Summanwar, V.S., Jayaraman, V.K., Kulkarni, B.D., Kusumakar, H.S., Gupta, K., & Rajesh, J. (2002). Solution of constrained optimization problems by multi-objective genetic algorithm. *Computers and Chemical Engineering*, 26, 1481–1492.

- Wang, R.C., & Fang, H.H. (2001). Aggregate production planning with multiple objectives in a fuzzy environment. *European Journal of Operational Research*, 133, 521–536.
- Wang, R.C., & Liang, T.F. (2004). Application of fuzzy multi-objective linear programming to aggregate production planning. *Computers and Industrial Engineering*, 46 (1), 17–41.
- Wang, R.C., & Liang, T.F. (2005). Applying possibilistic linear programming to aggregate production planning. *International Journal of Production Economics*, 98, 328-341.
- Yeh, W.C., & Chuang, M.C. (2011). Using multi-objective genetic algorithm for partner selection in green supply chain problems. *Expert Systems with Applications*, 38, 4244-4253.
- Zadeh, L.A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1, 3–28.
- Zimmermann, H.J. (1976). Description and optimization of fuzzy systems. *International Journal of General Systems*, 2, 209–215.
- Zimmermann, H.J. (1978). Fuzzy programming and linear programming with several objective functions. *Fuzzy Sets and Systems*, 1, 45–56.