Optimization of a Bi-functional APP Problem by using multi-objective genetic algorithm (NSGA-II)

Anwar sadath M A[1], Manoj P J[2]

[1] *Assistant Professor, Department of mechanical Engineering, ICET Mulavoor, Kerala, India*
[2] *Associate Professor, Department of mechanical Engineering, GEC Thrissur, Kerala, India*

Abstract - *Optimization of a problem having multiple and conflicting objectives were always a difficult task for decision makers. Aggregate Production Planning (APP) was always such a problem for mangers, which when studied carefully produces a variety of conflicting objectives to optimize simultaneously. This paper proposes the usage of a fast and elitist multi objective genetic algorithm, NSGA-II to optimize a multi-product, multi-Period APP Problem. The multiple objectives considered under study are Maximization of Sales Revenue, Minimization of total cost and minimization of inventory cost for a limited storage facility condition. The model proposed is successfully implemented using MATLAB Software.*

Key Words: APP problem, Optimization, Genetic Algorithm, NSGA, NSGA-II.

1. INTRODUCTION

Aggregate production planning is a mid-term planning process concerned with the determination of production, inventory, and work force levels to meet fluctuating demand requirements over a planning horizon that ranges from six months to one year. A planner must make decisions according to the optimal combination of production rate, work force level and inventory level over the planning horizon to optimize the production plan. Achieving a balance of expected supply and demand is the goal of aggregate planning. The APP problem deals with how to employ the available workforce, resources and facilities, including external contractors, to best satisfy the demand which is defined through APP [4].

1.1 MULTI OBJECTIVE OPTIMIZATION

The presence of multiple objectives in a problem, in principle, gives rises to not only single optimal solution but a set of optimal solutions (largely known as Pareto-optimal solutions). Pareto-optimal solutions or non dominated solutions are the set of solutions which are superior to the rest of solutions in the search space when all objectives are considered. This Solution set is produced by making tradeoff between the objectives. Since none of the solutions in the nondominated set is absolutely better than any other, any one of them is an acceptable solution. One way to solve multiobjective problems is to scalarize the vector of objectives into one objective by averaging the objectives with weight vector. This process allows a simpler optimization algorithm to be used, but the obtained solution largely depends on the weight vector used in the scalarization process. Moreover this method does not provide any insight to alternate solutions, if any to the decision maker. To overcome these drawbacks Genetic algorithms are considered since its ability to work with a population of points, which can capture a number of pareto-optimal solutions.

1.2 NSGA-II

The Objective of the study is to produce a Pareto Optimal Solution set for the Multi Product Multi Period APP Problem using Non Dominated Sorting based Genetic Algorithm NSGA-II. NSGA-II is the second version of the famous "Non-dominated Sorting Genetic Algorithm" based on the work of Prof. Kalyanmoy Deb of Kanpur Genetic Lab, for solving non-convex and non-smooth single and multi-objective optimization problems [7].

Its main features are:

- A non-dominated sorting procedure where all the individual are sorted according to the level of non-domination;
- It implements elitism which stores all non-dominated solutions, and hence enhancing convergence properties;
- It adapts a suitable automatic mechanics based on the crowding distance in order to guarantee diversity and spread of solutions.

The goal of this paper is to formulate an APP problem as a multi-objective optimization and illustrate its solution using Pareto based multi-objective optimization NSGA-II. The APP initialization and the NSGA-II optimization are implemented using MATLAB Software.

2. PROBLEM DISCRIPTION

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 Non dominated Sorting based Genetic Algorithm (NSGA-II) approach to determine the optimum aggregate plan for meeting forecasted demand by adjusting regular and overtime production rates, inventory levels, labor levels, subcontracting 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.

1) The values of all parameters are certain over the next *T* planning horizon.
2) The escalating factors in each of the costs categories are certain over the next *T* planning horizon.
3) Actual labor levels, and warehouse space in each period cannot exceed their respective maximum levels.
4) The forecasted demand over a particular period have to be satisfied, backorder is not entertained.

In this study, a multi product production planning problem faced by a crumb rubber production unit, Alwaye Techno Rubbers Pvt Ltd., Ernakulum is investigated. The company produces 4 different grades of crumb rubber used in tyre manufacturing. The APP problem under study has 3 objective functions. Maximization of Sales Revenue, Minimization of total cost and minimization of inventory cost.

2.1 MATHEMATICAL NOTATIONS & PARAMETERS

Notations

i = No of Products, i=1,2,3,..

t = No of periods in the planning horizon, j=1,2,3,..

Input parameters

S_{it} = Sale price (per ton) of product *i* at period *t*

D_{it} = Demand of product *i* at period *t*

P_{it} = Quantity of product *i* manufactured at normal working hours at period *t*

Cp_{it} = Production cost for manufacturing product *i* at normal working hours at period *t*

Po_{it} = Quantity of product *i* manufactured at overtime working hours at period *t*

Cpo_{it} = Production cost for manufacturing product *i* at overtime working hours at period *t*

I_{it} = Quantity of product *i* at inventory during period *t*

Ci_{it} = Inventory cost for storing product *i* at period *t*

W_{it} = Work force employed to produce product *i* at period *t*

Cp_{it} = Cost per worker for producing product *i* at period *t*

Psb_{it} = Quantity of product *i* manufactured by subcontracting at period *t*

Csb_{it} = Subcontracting cost for manufacturing product *i* at period *t*

I_{it-1} = Quantity of product *i* at inventory during period *t-1*

2.2 OBJECTIVE FUNCTIONS

This model contains three objectives

1) Maximization of total sales revenue (Z_1)
2) Minimization of total cost (Z_2) and

$$Max\ Z_1 = \sum_{i=1}^{N} \sum_{t=1}^{T} S_{it} D_{it}$$

$$Min\ Z_2 = \sum_{i=1}^{N} \sum_{t=1}^{T} P_{it} Cp_{it} + \sum_{i=1}^{N} \sum_{t=1}^{T} Po_{it} Cpo_{it} + \sum_{i=1}^{N} \sum_{t=1}^{T} I_{it} Ci_{it} + \sum_{i=1}^{N} \sum_{t=1}^{T}$$

2.3 CONSTRAINTS

(1) Demand Constraint
$Dit_{min} \le Dit \le Dit_{max}$

(2) Production limit constraints for each product
$P_{it\,min} \le P_{it} + P_{oit} \le P_{it\,max}$
The sum of normal time production and overtime production of each item should be between the minimum and maximum production limit.

(3) Total Workers Constaint
$Wt_{min} \le W_t \le Wt_{max}$

(4) Overtime Workers Constraint
$Wot_{min} \le W_t \le Wot_{max}$

(5) Inventory Constraint
$I_{it\,min} \le P_{it} + Po_{it} + I_{it-1} - D_{it} \le I_{it\,max}$

3. METHADOLOGY

The formulated model is to be solved by Non Dominated Sorting based Genetic Algorithm-II (NSGA-II) developed by Dr. Kalyanmoy Deb and team, at Kanpur Genetic Algorithms Laboratory. It is the updation and second version of the famous "NSGA" algorithm by Dr. Kalyanmoy Deb himself for solving non-convex and non-smooth single and multiobjective optimization problems.

NSGA suffers from three weaknesses, computational complexity, non-elitist approach and the need to specify a sharing parameter [8]. NSGA-II resolved the above problems and uses elitism to create a diverse Pareto-optimal front. The main features of NSGA-II are low computational complexity, parameter less diversity preservation, elitism and real valued representation.

NSGA-II implements elitism for multi-objective search, using an elitism-preserving approach. Elitism is introduced by storing all non-dominated solutions discovered so far, beginning from the initial population. Elitism enhances the convergence properties towards the Pareto-optimal set. A parameter-less diversity preservation mechanism is adopted. Diversity and spread of solutions are guaranteed without the use of sharing parameters, since NSGA-II adopts a suitable parameter-less niching approach. It uses the crowding distance, which estimates the density of solutions in the objective space, and the crowded comparison operator, which guides the selection process towards a uniformly spread Pareto-frontier.

3.1 NON DOMINATION

A solution is called nondominated, or Pareto optimal, if none of the objective functions can be improved in value without degrading some of the other objective values. Non domination can be better explained by the figure 3.1.

3.1.1 Domination:
One Solution is said to dominate another if it is better in all objectives

3.1.2 Non-Domination [Pareto Points]:
A solution is said to be non dominated if it is better than other solutions in at least one objective.

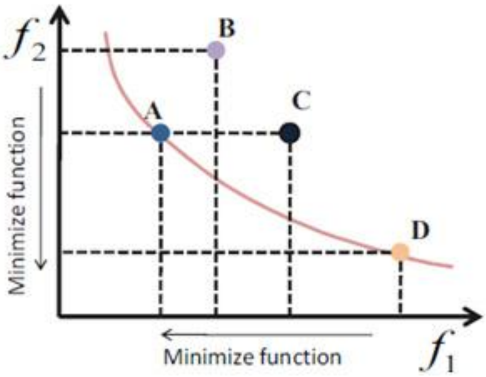


Figure 3.1: Non domination Pareto front

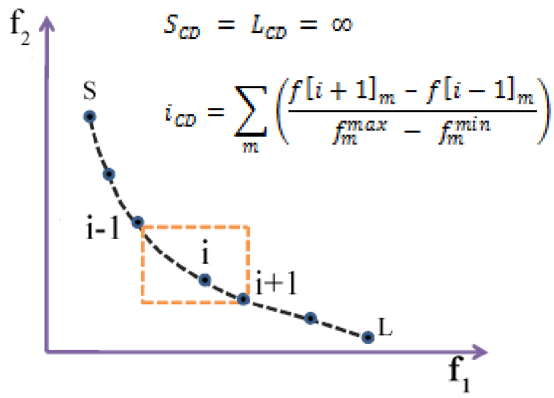
- A dominates B (better in both f1 and f2)
- A dominates C (Same in F1 but better in f2)
- A does not dominate D (non dominated points)
- A and D are in Pareto Optimal Front
- These non dominated solutions are called Pareto optimal Solutions
- This non dominated curve is called Pareto front

3.2 DIVERSITY MECHANISM BASED ON CROWDING DISTANCE

Crowding distance assignment helps to get an estimate of density of solutions surrounding a particular solution in population. Choosing individuals having large crowding distance ensures diversity in solution space.

The crowding-distance computation requires sorting the population according to each objective function value in ascending order of magnitude. Thereafter, for each objective function, the boundary solutions (solutions with smallest and largest function values) are assigned an infinite distance value. All other intermediate solutions are assigned a distance value equal to the absolute normalized difference in the function values of two adjacent solutions.

To get an estimate of the density of solutions surrounding a particular solution in the population, we calculate the average distance of two points on either side of this point along each of the objectives. This quantity serves as an estimate of the perimeter of the cuboid formed by using the nearest neighbors as the vertices (call this the *crowding distance*). In Figure 3.2, the crowding distance of the i^{th} solution in its front (marked with solid circles) is the average side length of the cuboid (shown with a



dashed box).

Figure 3.2: Crowding distance

3.3 GENETIC OPERATORS.

Genetic algorithm (GA) is a search heuristic that mimics the process of natural selection. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover.

In genetic algorithms, crossover is a genetic operator used to vary the programming of a chromosome or chromosomes from one generation to the next. It is analogous to reproduction and biological crossover, upon which genetic algorithms are based. Cross over is a process of taking more than one parent solutions and producing a child solution from them.

Mutation is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. Mutation alters one or more gene values in a chromosome from its initial state. In mutation, the solution may change entirely from the previous solution. Hence GA can come to better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search.

NSGA-II uses Simulated Binary Crossover (SBX) [19] operator for crossover and polynomial mutation as mutation Operator.

3.4 NSGA-II PROCEDURE

In NSGA-II, the offspring population Qt is first created by using the parent population Pt, of size N. However, instead of finding the nondominated front of Qt, the two populations are combined together to form Rt of size 2N. This implements elitism in the process. Then, non-dominated sorting is used to classify the entire population Rt. The new population is filled by solutions of different nondominated fronts, one at a time. The filling starts with the best non-dominated front and continues with solutions of the second non-dominated front, followed by the third, and so on. Since the overall population size of Rt is 2N, not all fronts may be accommodated in N slots available in the new population. All fronts which could not be accommodated are simply deleted. When the last allowed front is being considered, there may exist more solutions in the last front than the remaining slots in the new population. Instead of arbitrarily discarding some members from the last front, a niching strategy, 'crowding distance' is used to choose the members from the last front, which reside in the least crowded region in the front. The algorithm ensures that niching will choose a diverse set of solutions from this set. When the entire population converges to the Pareto-optimal front, the continuation of this algorithm will ensure a better spread among the solutions. The schematic representation of NSGA-II procedure is shown in Figure 3.3.

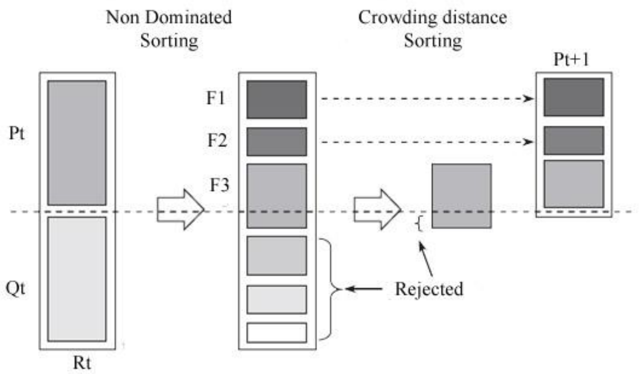


Figure 3.3: NSGA-II Procedure.

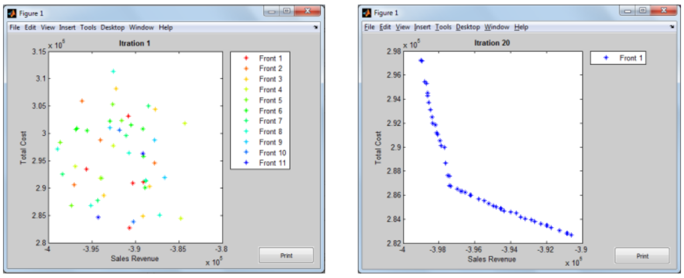
4. RESULTS AND DISCUSSIONS

The APP Problem under consideration has two objective functions to optimize simultaneously. Bi-Functional Optimization is easy to understand and analyze since it only uses two objective functions and the convexality of the pareto front is easily recognizable in the graph. Here Bi-objective optimization is performed entirely to confirm the convexality of the solution space and there by the success of the NSGA-II implementation. The objective functions selected are

Z_1 = Max. Sales Revenue and

Z_2= Min. Total Cost

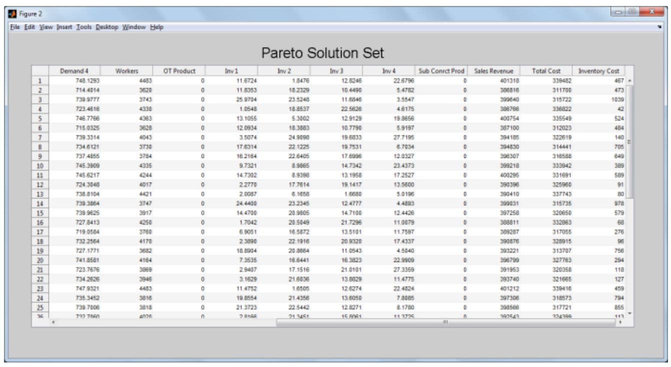
The Bi-Functional Optimization produced a perfect Convex Pareto Front, Which can be analyzed below.



(a) Initial Population (b) Final Pareto Front

Figure 4.1: Bi-Functional Pareto Front

Form figure 4.1(a) we can see that the initial population is a randomly spread all over the solution space, and these random solutions are separated in to different fronts represented by different colours in the figure 4.1(a) by using a non dominated sorting mechanism. During the program running, we can see that, with each iteration the solution space is confined to a lesser number of fronts and the shape of the solution space are rearranged into a convex form. By using non dominated sorting it is found that even the complex problem can produce a single front solution space within 10 to 20 iterations. The figure 4.1(b) shows the final pareto front produced after 20 iterations. From the figure we can see that it has a perfect convex shape specifying a perfect optimal Pareto front. We can also analyze that the diversification strategy presented by the NSGA-II, produces a perfect uniformly distributed solution set between the upper and lower bound elements. This is the most interest case for decision makers. When the Pareto front has this shape, the decision makers can negotiate, fighting for their own objective and they can more easily agree for a trade-off point. In this situation, the trade-off is much better than the linear combination of the original objectives. This means, practically, that if a decision maker gives up a percentage of its target, say 20%, another decision maker may have an improvement of more than 20% on his personal target.



Pareto Solution Set

	Demand 4	Workers	OT Product	Inv 1	Inv 2	Inv 3	Inv 4	Sub Conrct Prod	Sales Revenue	Total Cost	Inventory Cost
1	748.1293	4483	0	11.6724	1.8476	12.8246	22.6796	0	401318	339482	467
2	714.4014	3620	0	11.8353	18.2329	10.4490	5.4782	0	386816	311700	473
3	739.9777	3743	0	25.9704	23.5248	11.6846	3.5547	0	399640	315722	1039
4	723.4616	4330	0	1.0548	18.8537	22.5626	4.8175	0	388768	336822	42
5	746.7766	4363	0	13.1055	5.3802	12.9129	19.8656	0	400754	335549	524
6	715.0325	3628	0	12.0934	18.3883	10.7790	5.9197	0	387100	312023	484
7	739.3314	4043	0	3.5074	24.9090	19.6833	27.7195	0	394185	322619	140
8	734.6121	3730	0	17.6314	22.1225	19.7531	6.7034	0	394830	314441	705
9	737.4855	3784	0	16.2164	22.6405	17.6996	12.0327	0	396307	316588	649
10	745.3909	4335	0	9.7321	8.9865	14.7342	23.4373	0	399218	333942	389
11	745.6217	4244	0	14.7302	8.9398	13.1958	17.2527	0	400295	331691	589
12	724.3048	4017	0	2.2770	17.7614	19.1417	13.5600	0	390396	325960	91
13	738.8104	4421	0	2.0087	6.1658	1.6680	5.0196	0	390410	337743	80
14	739.3884	3747	0	24.4400	23.2345	12.4777	4.4893	0	399031	315735	978
15	739.9625	3917	0	14.4700	20.9805	14.7100	12.4426	0	397258	320650	579
16	727.8413	4250	0	1.7042	20.5049	21.7296	11.0879	0	388811	332863	68
17	719.0584	3760	0	6.9051	16.5872	13.5101	11.7597	0	389287	317055	276
18	732.2564	4170	0	2.3890	22.1916	20.9320	17.4337	0	390876	328915	96
19	727.1771	3682	0	18.8904	20.8664	11.0543	4.5040	0	393221	313707	758
20	741.8581	4164	0	7.3535	16.6441	16.3823	22.9909	0	396799	327783	294
21	723.7676	3869	0	2.9407	17.1516	21.0101	27.3359	0	391953	320358	118
22	734.2626	3946	0	3.1629	21.6036	13.8829	11.4775	0	393740	321665	127
23	747.9321	4483	0	11.4752	1.6505	12.6274	22.4824	0	401212	339416	459
24	735.3452	3816	0	19.8554	21.4356	13.6050	7.8085	0	397308	318573	794
25	739.7006	3818	0	21.3723	22.5442	12.8271	8.1780	0	398566	317721	855

Figure 4.2: Pareto Solution Set

The Tabular Analysis provided by the program gives all the optimum condition values of the pareto solution set. which includs all the input variables like demand and production of each item, Inventory levels of each item etc, and the output variables Sales Revenue, Total cost and Inventory cost. This pareto solution set is produced with different kind of tradeoffs between the objective functions. Now the decision maker has to choose from this solution set an optimum condition suited for his work condition.

5. CONCLUSIONS

For a multi product, multi period APP Problem, Pareto front Solution Space is achieved correctly with NSGA-II implementation in MATLAB. The solution space for the Bi-Functional Optimization is studied graphically and analytically. The graphical analysis of the bi-functional optimization shows a perfect convex shaped pareto front, signifying the success of NSGA-II implementation for multi objective optimization of the APP problem. The number of iterations required to reach a single front pareto solution set by using NSGA-II is found out to be very less. It also produced a uniform distributed solution space. The non dominated ranking, crowded tournament selection and the elitism used by the NSGA-II produced these better results.

REFERENCES

[1]. 'Solving an aggregate production planning problem by using multi-objective genetic algorithm (MOGA) approach', Ripon Kumar Chakrabortty and Md. A. Akhtar Hasin, International Journal of Industrial Engineering Computations 4 (2013)

[2]. Monim A. Gasim, "Aggregate Production Planning Using Goals Programming", Al-Rafidain Engineering Vol.21 No. 3 June 2013.

[3]. Stephen c. H. Leung, yue wu and k. K. Lai, "multi-site aggregate production planning with Multiple objectives: a goal programming approach", production planning & control, vol. 14, no. 5, july–august 2003

[4]. B. Fahimnia, L.H.S. Luong, and R. M. Marian, "Modeling and Optimization of Aggregate Production Planning - A Genetic Algorithm Approach", International Journal of Applied Mathematics and Computer Sciences Volume 1 Number 1

[5]. Lorena PRADENAS, C_esar ALVAREZ, and Jacques A. FERLAND," A Solution for the Aggregate Production Planning Problem in a Multi-Plant, Multi-Period and Multi-Product Environment"

[6]. Maciej nowak, "An interactive procedure for aggregate production planning", croatian operational research review (CRORR), vol. 4, 2013

[7]. Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan, A Fast Elitist Multiobjective Genetic Algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6 (2002), no. 2, 182 ~ 197.

[8]. N. Srinivas and Kalyanmoy Deb, Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms, Evolutionary Computation 2 (1994), no. 3, 221 ~ 248.

[9]. Meredith, J. R. & Shafer, S. M. "Operations Management for MBAS", John Wiley & Sons Inc., New York, 2001.

[10]. Wiley, Chichester, Multi-objective Optimization using Evolutionary Algorithms, UK, 2001

[11]. C. M. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," in Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest, Ed. San Mateo, CA: Morgan Kauffman, 1993, pp. 416–423.

