Optimal Aggregate Production Planning by using Genetic Algorithm

Poonam Savsani *(Author)*
Assistance Professor, Department of Industrial Engineering
Pandit Deendayal Petroleum University
Gandhinagar, Gujarat, India
poonam.savsani@sot.pdpu.ac.in

Gaurav Banthia *(Author)*
Student, Department of Industrial Engineering
Pandit Deendayal Petroleum University
Gandhinagar, Gujarat, India
gaurav.bie12@sot.pdpu.ac.in

Juhi Gupta *(Author)*
Student, Department of Industrial Engineering
Pandit Deendayal Petroleum University
Gandhinagar, Gujarat, India
juhi.gie12@sot.pdpu.ac.in

Ronak Vyas *(Author)*
Student, Department of Industrial Engineering
Pandit Deendayal Petroleum University
Gandhinagar, Gujarat, India
ronak.vie12@sot.pdpu.ac.in

***Abstract*—Aggregate production planning (APP) deals with the simultaneous determination of plant's production, inventory and vocation levels over a finite time horizon. The aim of aggregate planning is to finalize overall output levels in the near to medium future in uncertain demands. This paper presents a Genetic Algorithm approach for solving aggregate production planning with different selection methods and various crossover phenomenon. Combination of four selection methods and five crossover phenomenon are taken and compared to choose the best combination for solving APP in this present work. The problem statement depicts multi-product, multi-period APP with forecasted demand. The proposed approach attempts to minimize the total cost which includes labor cost, backordering cost, subcontracting cost, inventory cost, warehouse cost, overtime cost and machine cost. Results show the outstanding performance of uniform selection procedure and two point crossover combination.**

Keywords*—*Aggregate production planning, genetic algorithm.

1. Introduction

Aggregate production planning is making a production plan for a product, 6 to 18 months in advance. In today's world where everything is so unstable and unpredictable we cannot make a fixed production plan for a product. It may happen that we assume that our sales will hit high during festivals and we produce more products but they might not be sold. This might be because our product is not a seasonal product and it is sold equally throughout the year or because it is required days before a

festival. For example, we produce cycle tyres. We have a very good market for cycle during the festival of Dussehra but in order to sell cycles on the day of Dussehra we need to make them about 1 month in advance. Now because arranging a large amount of raw materials is also a hurdle, we need to make sure we do not face any delays. In order to avoid delays, we should make our production plan accordingly but if we have a fixed production plan then this might not be possible. Thus here we need a flexible type of production planning which can also be known as Aggregate Production Planning (APP).

There are many advantage of APP, it provides an idea to management as to what quantity of materials and other resources are to be procured and when, so that the total cost of operations of the organization is kept to the minimum over that period. The quantity of outsourcing, subcontracting of items, overtime of labour, numbers to be hired and fired in each period and the amount of inventory to be held in stock and to be backlogged for each period are decided. All of these activities are done within the framework of the company ethics, policies, and long term commitment to the society, community and the country of operation. (Ripon Kumar Chakrabortty & Md. A. Akhtar Hasin, 2012).

APP has attracted significant interest from both practitioners and academics. For solving APP problems, certain constraints are imposed which demand constraint optimization. The genetic algorithm introduced by Ioannis (2009) which is about the problem of constrained optimization and he also came up with improved version of genetic operators i.e. crossover and mutation. This improved version conserves the feasibility of the trial solutions of the constrained problem that are encoded in the chromosomes. Bunnag and Sun (2005) also emerged with the real coded Genetic Algorithm which is based on stochastic (probability based) optimization method, referred to as a Genetic Algorithm (GA), for solving constrained optimization problems over a compact search domain. This converges in probability to the optimal solution by treating through a repair operator.

Genetic Algorithm (GA) is known for providing appreciable amount of alternative solutions for various GA parameter values. The early evolution of Genetic Algorithm can be chased from late 1950's and early-1960's (Bremermann, H.J., 1958). Evolutionary biologist programmed it on computer and was explicitly seeking to the model aspects of natural evolution.

By 1962, researchers like G.E.P Box, G.J Friedman, W.W Bledsoe and H J Bermermann emerged with the work on function optimization and machine learning but their work did not popularized. In the same year John Hollands came up and laid foundation on Adaptive Systems and got recognized. He was also the first to propose crossover and other recombination's operations (J. H. Holland, 1962). In the recent years many new such effective optimization methods caught the attention of different researchers such as teaching learning based optimization, biogeography based optimization, cuckoo search algorithm, ant colony optimization, Passing vehicle search etc. (Savsani et al., 2013, 2014a, 2014b, 2015).

Ingo Rechenberg also introduced a more successful development in this area (student of the Technical University of Berlin), called Evolution Strategy, though it was more similar to hill climbers than to GA. In this technique, he presented that one parent was mutated to produce one off spring and the other better half of the two was kept and become the parent for the next round of mutation. He also stated that there was more no population or crossovers. Later versions introduced the idea of populations (I. Rechenberg, 1965).

The next important development in the field came in 1966, when Fogel, L.J, A.J Owens & M.J Walsh introduce in America a technique they called Evolutionary Programming. In this technique, a candidate solution to the problem was represented as simple finite state machines like Rechenberg's evolution strategy. Their algorithm worked by randomly mutating one of the simulations machines and keeping the better of the two (Fogel et al., 1966).

Genetic algorithms in particular became popular through the work of John Holland in the early-1970. In 1975, Holland and few of his colleagues at the University of Michigan were first to systematically and rigorously present the concept of adaptive digital systems using mutation, selection and crossovers (J. H. Holland, 1975). Simulating process of biology evolution as a problem solving strategy. In the same year, Kenneth De Jong's important dissertation established the potentials of GA by showing that they could perform very well on a wide variety of test functions including noisy, discontinues and multimodal search landscape.

By mid-1980's GA was being applied to s broad range of subjects from abstract mathematical problem like bin packing, graph coloring to tangible engineering issues such as pipeline flow control, pattern recognition (E. Pettit and K.M. Swigger, 1983) & classification and structural optimization (D. E. Goldberg, 1989).

Today GA's are solving problems of everyday interests, in areas of study as diverse as stock market prediction and portfolios planning, aerospace engineering, micro-chip design, biochemistry and molecular biology and scheduling at airports and assemble lines.

GENETIC ALGORITHM

A genetic algorithm (GA) is a method for solving both constrained and unconstrained optimization problems based on a natural selection process that mimics biological evolution just as the living being on earth adapt themselves to any changes in the environment for survival, if they don't they die. It is search heuristic which search for the best solution to the problem. In this paper, population of 20 candidates (parents) is taken in which each candidate solution has its own set of properties (its chromosomes or genotype) which can be mutated and altered and follow the biological evolution. These Chromosomes and genotypes are in the array called strings which are in the form of binary digits i.e. 0 and 1. The more fit individuals (0 or 1) are stochastically selected from the current population. Then Crossover is then applied on these strings containing Chromosomes forming new strings with more fitness, also called Offspring or the next generation. This is followed by Mutation which does random alteration just to create diversity.

Each new solution tries to achieve its best possible fitness and pass the same best solution to the new generation. This is how each new offspring is evolved, through various iteration (generation), with more fitness to its predecessor and more forward towards to the minima (minimal deviation from fitness). This fitness is achieved by the three operator- selection, crossover and mutation. The fitness is usually the value of the objective function in the optimization problem being solved. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. In other words, it is terminated when best possible fit is achieved and that is optimized solution.

Outline of the Basic Genetic Algorithm (Marek Obitko)

i. ***[Start]*** Take the population of 20 chromosomes.
ii. ***[Fitness]*** Compute the value of the fitness function f(x) for each chromosome in the population.
   a. ***[New population]*** Through a following steps and various iteration, a new population will emerged called Offspring.
   b. ***[Selection]*** It is the selection based on the fitness of two parent chromosomes. Parent's chromosomes with the better fitness have more chances to be selected.
   c. ***[Crossover]*** It is process of producing offspring which crossover the parents. It is done to improve the fitness of the new generation and to avoid exact copy of parents.
   d. ***[Mutation]*** It allows the random modification.
iii. ***[Accepting]*** Accept the resultant new offspring.
iv. ***[Replace]*** Replace the old chromosomes with new ones and use this to find the generation with better fitness.
v. ***[Test]*** Terminate if the optimal condition is acquired.

[Loop] Go to step 2

2. Problem Statement

2.1 Problem Statement and notations

The portraiture of the multi-product APP problem is as follows and taken as the problem statement for the paper. Here, the assumption is taken for the planning horizon t, over which the company manufactures N kinds of products to meet the market demand of the people. The solution to this APP problem is evaluated using GA Algorithm which gives optimum levels of labor, inventory, backordering and subcontracting rates, overtime and regular production rates and other controllable variables. The mathematical model is developed taking some assumptions, based on the characteristics of the APP problem.

i. The values of all the parameter are obsolete and known.
ii. The escalating factors for each category of cost are invariant.
iii. Actual Machine capacity, Labor levels and warehouse space in each period should not exceed their maximum levels.
iv. Every forecasted demand can be either satisfied or backordered in a particular period, but the backorder will be fulfilled in the next period.

The above provided problem statement and following notation are used after reviewing the literature and considering practical situations (Wang & Liang, 2004; Masud & Hwang, 1980; Wang & Fang, 2001; Ripon Kumar Chakrabortty & Md.A.Akhtar, 2013).

D_{nt}: Forecasted demand for nth product in period t (units)
A_{nt}: Regular time production cost per unit for nth product in period t (Tk. /unit)
Q_{nt}: Regular time production for nth product in period t (units)
i_a: Escalating factor for regular time production cost (%)
b_{nt}: Overtime production cost per unit for nth product in period t (Tk. /unit)
O_{nt}: Overtime production for nth product in period t (units)
i_b: Escalating factor for overtime production cost (%)
c_{nt}: Subcontracting cost per unit of nth product in period t (Tk. /unit)
S_{nt}: Subcontracting volume for nth product in period t (units)
i_c: Escalating factor for subcontract cost (%)
d_{nt}: Inventory carrying cost per unit of nth product in period t (Tk. /unit)
I_{nt}: Inventory level in period t for nth product (units)
i_d: Escalating factor for inventory carrying cost (%)
e_{nt}: Backorder cost per unit of nth product in period t (Tk. /unit)
B_{nt}: Backorder level for nth 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 nth product in period t (man-hour/unit)
r_{nt}: Hours of machine usage per unit of nth product in period t (machine-hour/unit)
V_{nt}: Warehouse spaces per unit of nth product in period t (feet2/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 (feet2)

2.2 Single Objective Genetic Algorithm Model

2.2.1 Single Objective Function

Most practical decisions made to solve APP problems usually consider total costs. The proposed GA Algorithm targeted objective function. 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:

$$MinZ_1 = \sum_{n=1}^{N}\sum_{t=1}^{T}[a_{nt}Q_{nt}(1+i_a)^t + b_{nt}O_{nt}(1+i_a)^t + c_{nt}S_{nt}(1+i_c)^t + d_{nt}I_{nt}(1+i_d)^t + e_{nt}B_{nt}(1+i_e)^t]$$
$$+ \sum_{t=1}^{T}(K_tH_t + m_tF_t)(1+i_f)^t$$

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.

2.2.2 Constraints

Constraint on Carrying Inventory

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

$$I_{nt} \geq I_{nt\,\min} \quad for \forall n \forall t \tag{2}$$

$$B_{nt} \leq B_{nt\,\max} \quad for \forall n \forall t \tag{3}$$

Where, D_{nt} denotes the imprecise forecast demand of nth product in period t.

Constraint on Labor Levels

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

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

Constraint on Machine Capacity and Warehouse Space

$$S_{nt} \leq S_{nt\,\max} \quad for \forall n, \forall t \tag{6}$$

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

$$\sum_{n=1}^{N} V_{nt} I_{nt} \leq V_{t\,\max} \quad for \forall t \tag{8}$$

Non Negativity Constraint

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

2.3 Genetic Algorithm Parameters

2.3.1 Crossover Options: It explains how the Genetic Algorithm helps to form a crossover child from two individual or two parents.

1. Scattered Crossover: The some genes are collected from parent 1 and rests are collected from parent 2. It creates a random binary vector by selecting genes where vector is 1 from parent 1 and where vector 0 is from parent 2. It combines together to form a child. E.g.: Let us suppose parent p1 have [q w e r t a] and p2 have [1 2 3 4 5 6], and the binary vector is [1 0 0 0 1 1]. Then, the child would have [q 2 3 4 t a].

2. Single Point Crossover: It selects a random integer from total number of variables. It selects the vector entries equal to or less than from parent 1 and rest are fill up by parent 2. E.g.: Let us suppose parent p1 have [q w e r t a] and parent 2 have [1 2 3 4 5 6] and the random integer is 2 then, the child would have [q w 3 4 5 6].

3. Two Points Crossover: It selects two random integers let say 'a' and 'b' instead of one, from total number of variables (m). The vector entries from 1 to a will be from parent 1, vector entries from a+1 to b will be from parent 2 and vector entries from b+1 to m will be again from parent 1. E.g.: Let us suppose parent p1 have [q w e r t a] and parent p2 have [1 2 3 4 5 6] and the random values are 2 and 4. Then, child would have [q w 3 4 t a].

4. Arithmetic Crossover: It helps to produce two new offspring's after combining linearly chromosomes of both the parents.

   E.g.: Offspring1=z*parent (p1) + (1-z)*parent (p2)

   Offspring2= (1-z)*parent (p1) +z*parent (p2)

5. Intermediate Crossover: It creates children by random weighted average of the parents. It is controlled by only one parameter named RATIO.

   E.g.: Offspring1=x*parent (p1) + (1-x)*parent (p2)

   Offspring2= (1-x)*parent (p1) + x*parent (p2)

   x= (1+2*c) r-c c=exploration coefficient-user defined (c>=0)

   r=random number between 0 and 1.

2.3.2 Selection Options: It chooses the parents for the next generation based on their scaled values from fitness scaling functions.

1. Tournament Selection: It selects individual from the population. It involves running tournaments between the selected chromosomes from the population. And the winner of each tournament is selected for crossover. If tournament is larger, weak individual have a smaller chance to get selected.

2. Roulette Selection: It is kind of roulette game. It stimulates a roulette wheel with the area of each segment proportional to its expectation. The algorithm then uses a random number to select one of the sections with a probability equal to its area.

3. Uniform Selection: It selects parent at random from a uniform distribution using the expectations and number of parents. This results in an undirected search. It is not a useful search strategy but can be used to test genetic algorithm.

4. Remainder Selection: It assigns parents deterministically from the integer part of each individual's scaled value and then uses roulette selection on the remaining fractional part.

The following case has been taken after reviewed the practical situation (Ripon Kumar Chakrabortty & Md.A.Akhtar, 2013)

3. Model Implementation

3.1 Case Description

This paper is confronted with the of case of the Comfit Composite Knit Limited (CCKL),a Ready Made Garments manufacturing company and a 100% export oriented composite knit textile unit which established in Bangladesh with the allegiance to serve the Global needs for knit and casual clothing. This has been taken to manifest as a model for the proposed methodology. Among all, the main focus will on two expensive product of the company i.e. Hooded Jackets(product 1) and Ladies Cardigan(product 2).Since these are most expensive and are heavy in volume, significant amount of time and cost of all manufacturing items are needed. Therefore, it needs a lot of meticulous observations and error-free manufacturing practices to manifest the market situation and to reduce time and cost to satisfy the customer. The case study presented here is in collaboration with Genetic Algorithm as one of the approach to minimize the total cost. The model has 2 months planning horizon i.e. May and June. According to the introductory information, Table 1 and Table 2 summarize the available resources, labor level, hiring and layoff cost and escalating factor. Operating cost data, forecasted demand, maximum labor, machine, warehouse capacity, back order level, subcontracted volume & minimum inventory data are given in table-3 and table-4. The population size, number of generations, and the number of runs which have been considered for the experimental run in MATLAB software for the above equations are 20, 100, and 20 respectively.

Table – 1 *Data of available resources for product-1 and product-2*

	PRODUCT-1	PRODUCT-2
Initial Inventory(period1)	500 Units	200 Units
End Inventory (period2)	300 Units	300 Units
Labour hours/unit	0.033 man-hour	0.05 man-hour
Machine hours/unit	0.18 machine-hour	0.08 machine-hour
Warehouse space/unit	1 square feet	1.5 square feet

Table – 2 *Data for labor level, hiring cost, layoff cost and escalating factor*

Initial Labour level	225 man-hour
Hiring cost	Tk.22/worker/hour
Layoff cost	Tk.8/worker/hour
Escalating factor	1%

Table – 3 *Related Operating Cost Data*

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

Table – 4 *Forecasted Demand, maximum labor, machine, warehouse capacity, back order level, subcontracted volume & minimum inventory data.*

Items (Product 1)	Period			Items (Product 2)	Period	
	1	2			1	2
D_{1t} (pieces)	1400	2875		S_{1tamx} (pieces)	200	350
D_{2t} (pieces)	1600	887		S_{2tamx} (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_{ttmax}(ft2)	1000	1000		B_{1tmax} (pieces)	200	600
				B_{2tmax} (pieces)	150	100

GA has different selection and crossover methods, like remainder selection, uniform selection, roulette selection, tournament selection, scattered crossover, single point crossover, two point crossover, intermediate crossover and arithmetic crossover. The purpose of this work is to check the effect of selection and crossover strategy on APP problem. Various combinations of crossover and selection procedures are tested for the APP problem. GADS MATLAB toolbox is implemented for the population size of -----and ----number of generations. Results are obtained for ----runs and compared based on different statistical values like best solution, mean solution, worst solution. The population size, number of generations, and the number of runs which have been considered for the experimental run in MATLAB software for the above equations are 20, 100, and 20 respectively.

4. Results and Findings

In the proposed work 20 different combinations of crossover and selection procedures are tested and results are obtained for ---runs. In single Objective Genetic Algorithm we came up with multiple solutions. Table-6 depicts the fitness value for all the combinations. It is observed from the results that combination of two point crossover and uniform selection procedure performs better than all other combination. Statistics of fitness value for the two point crossover and uniform selection procedure is given in table-7. Statistics values show that minimum cost is 237,360 Bangladeshi taka. Moreover no much variation is observed between best and mean fitness value. Based on these experiments initial plan for CCKL case can be suggested as per table-5. The population size, number of generations, and the number of runs which have been considered for the experimental run in MATLAB software for the above equations are 20, 100, and 20 respectively.

Table – 5 *Optimal values for product-1 and product-2 for CCKL case*

Items (Product 1)	Period			Items (Product 2)	Period	
	1	2			1	2
Q_{1t} (Units)	571.705	1049.435		Q_{2t} (Units)	674.495	454.635
O_{1t} (Units)	571.065	1055.655		O_{2t} (Units)	658.925	458.86
S_{1t} (Units)	185.22	326.41		S_{2t} (Units)	84.355	91.165
I_{1t} (Units)	438.585	557.015		I_{2t} (Units)	166.26	236.365
B_{1t} (Units)	8.795	572.245		B_{2t} (Units)	188.535	90.89
H_t (Man hours)	116.845	156.575		F_t (Man hours)	11.955	146.305

Table – 6 *Fitness value obtained by using different types of crossover*

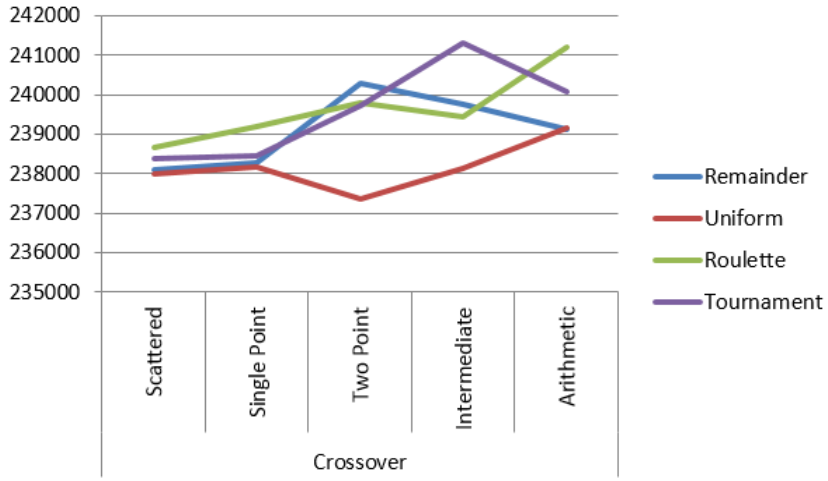
Selection	Crossover				
	Scattered	Single Point	Two Point	Intermediate	Arithmetic
Remainder	238085	238292	240292	239753	239110
Uniform	238001	238166	237360	238137	239164
Roulette	238662	239180	239791	239453	241198
Tournament	238366	238465	239712	241321	240063

Table – 7 *Statistics of the results obtained by using GA*

Mean	239129
Best Value	237360
Worst	241321
Tournament	1074.621

To check the behavior of different combination of crossover and selection strategies on APP problem, figure-1 is plotted. From Fig 1 several characteristics of this proposed approach can be drawn. Roulette selection procedure gives best performance with scattered crossover. The fitness value goes on increasing for combination of Roulette selection with single point crossover, two point crossover, intermediate crossover and arithmetic crossover. Tournament selection gives better performance with scattered crossover but performs worst with intermediate crossover. Remainder selection performs better with scattered crossover, getting worst with two point crossover and giving good results with arithmetic crossover. Uniform selection procedures gives best result with two point crossover. The population size, number of generations, and the number of runs which have been considered for the experimental run in MATLAB software for the above equations are 20, 100, and 20 respectively.

Figure – 1 *Variation of fitness value by using different crossover and selection strategies*



5. Conclusion

The proposed Single Objective Genetic Algorithm approach can solve the real life APP problem. This APP model can easily expanded by adding parameters, decision variables and constraints as required for practical use in industries. For the optimization of APP problems parameter of GA are rarely investigated. Therefore, in this paper, combination of five different types of crossover and four different types of selection procedures are experimented to optimize real life APP problem with the forecast demand, related operating costs, and capacity. The results reveal better performance of uniform selection procedure with two point crossover. Moreover scattered crossover gives near optimal fitness value with all the selection procedures. Among all these 20 combinations worst is the intermediate crossover with tournament selection procedure for the APP problem.

6. References

[1] Bremermann, H.J. “The evolution of intelligence. The nervous system as a model of its environment”, Technical report, no.1, contract no. 477(17), Dept. Mathematics, Univ. Washington, Seattle, July, 1958.

[2] Bunnag, D., & Sun, M. (2005). Genetic algorithm for constrained global optimization in continuous variables. Applied Mathematics and Computation, 171, 604–636

[3] D. E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, 1989.

[4] E. Pettit and K.M. Swigger, “An analysis of genetic-based pattern tracking” in Pric. National Conf. on AI, AAAO ’83, pp. 327-332, 1983.

[5] Elfas Nino, Carlos Ardila, Alfredo Perez,Yezid Donoso “A Genetic Algorithm for Multiobjective Hard Scheduling Optimization”, Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844:Vol. V (2010), No. 5, pp. 825-836.

[6] Fogel, L.J, A.J Owens and M.J Walsh (1966), “Artificial Intelligence through Simulated Evolution”, New York, Wiley.

[7] Goldberg,D.E, "Optimization & Machine Learning, Genetic Algorithm in Search”, Pearson Education Pvt.Ltd.

[8] I. Rechenberg. “Cybernetic solution path of an experimental problem”. Royal Aircraft Establishment, Library translation NO. 1122, Farnborough, Hants., UK, August 1965.

[9] Ioannis, G.T. (2009). “Solving constrained optimization problems using a novel genetic algorithm”. Applied Mathematics and Computation, 208, 273–283.

[10] J. H. Holland. "Adaptation in natural and artificial systems". Ann Arbor: The University of Michigan Press, 1975.

[11] J. H. Holland. "Outline for a logical theory of adaptive systems". Journal of the Association for Computing Machinery, 3, 1962, pp. 297- 314.

[12] 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.

[13] 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.

[14] 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.

[15] Ripon Kumar Chakrabortty and Md.A.Akhtar, "Solving an aggregate production planning problem by using multi-objective genetic algorithm (MOGA) approach," International Journal of Industrial Engineering Computation4 (2013)1-12.

[16] 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.

[17] Vikas Gaba, Anshu Prashar, "Comparison of processor scheduling algorithms using Genetic Approach": International Journal of Advanced Research in Computer Science and Software Engineering: Volume 2. (8August 2012) URL: www.ijarcsse.com

[18] Wang, R.C and Fang, H.H. (2001). "Aggregate production planning with multiple objectives in a fuzzy environment". European Journal of Operational Research, 133, 521-536.

[19] Wang, R.C and Liang, T.F. (2004). "Application of fuzzy multi objective linear programming environment to aggregate production planning". Computers and Industrial Engineering, 46 (1), 17-41.

[20] Wang, R.C., & Liang, T.F. (2005). "Applying possibilistic linear programming to aggregate production planning". International Journal of Production Economics, 98, 328-341.

[21] MATLAB 2013 toolbox of genetic algorithm.

[22] P Savsani, RL Jhala, V Savsani, Applied Soft Computing 21, 542-553 (2014a). "Effect of hybridizing Biogeography-Based Optimization (BBO) technique with Artificial Immune Algorithm (AIA) and Ant Colony Optimization (ACO)"

[23] P Savsani, RL Jhala, VJ Savsani, Systems Conference (SysCon), 2013 IEEE International, 381-386 (2013)."Optimized Trajectory Planning of a Robotic Arm Using teaching learning based optimization (TLBO) and artificial bee colony (ABC) optimization techniques"

[24] P Savsani, RL Jhala, VJ Savsani, IEEE, Systems Journal, IEEE (Volume: PP, Issue: 99) pp 1-12 (2014b). "Comparative study of different metaheuristics for the trajectory planning of a robotic arm".

[25] Savsani, P., & Savsani, V. (2015). "Passing Vehicle Search (PVS): A novel metaheuristic algorithm. *Applied Mathematical Modelling"*.

7. Biography

Dr. Poonam Savsani, working as an assistant professor at Pandit Deendayal Petroleum University, Gujarat, India, in industrial engineering department. She has completed her Ph.D. in robot trajectory optimization using advanced meta-heuristics. Her research interest includes exploration of different optimization methods for the industrial problems, such as aggregate production planning problem, vehicle routing problem, cutting stock problem, network analysis, robot trajectory planning.

Gaurav Banthia, pursuing his B-tech degree in the department of Industrial Engineering from Pandit Deendayal Petroleum University, Gujarat, India. Currently, he is hounding in the arrays of BRTS (Bus Rapid Transit System, India) and has done his intern at KHS and Siemens. He is also a member of Tark – The Students Chapter of Industrial Engineering at Pandit Deendayal Petroleum University (PDPU) and American Society of Quality's Chapter at PDPU. His Interest includes Simulation, Supply chain Management and Quality.

Juhi Gupta, undertaking her B-tech degree in the department of Industrial Engineering from Pandit Deendayal Petroleum University, Gujarat, India. She is currently tracking her interest in Supply Chain management, Flexible manufacturing System and Facility Layout. She is also member of Tark – The Students Chapter of Industrial Engineering at Pandit Deendayal Petroleum University (PDPU) and American Society of Quality's Chapter at PDPU. She has also completed her intern in the companies like KHS and National Engineering Industries/National Bearing Company (NBC).

Ronak Vyas, currently pursuing his B-tech Degree in the department of Industrial Engineering from Pandit Deendayal Petroleum University, Gujarat, India. Prosecuting his interests in Heath Care industry and Banking sector. He has completed his 6 week industrial internship in Eicher Tractors, a subsidiary of TMTL. He has worked as a core committee member of Tark – The Students Chapter of Industrial Engineering at Pandit Deendayal Petroleum University (PDPU) and American Society of Quality's Chapter at PDPU.