A New Genetic Algorithm for Job Shop Scheduling

[1]Harmesh K.Vanmali, [2]Prof. V.S. Jorapur

[1,2]Dept. of Mechanical Engineering, FrCRCE, Bandra, Mumbai, Maharashtra, INDIA

Abstract

The Job-Shop Scheduling (JSS) can be defined as a planning of schedules with many variations according to the requirements. In Job-Shop Scheduling Problem (JSSP) environment, there are numbers jobs to be processed on numbers of machines with a certain objective function to be minimized or maximized. In this paper, we have used the GT-GA to solve the job shop scheduling to minimize the makespan along with the special type of crossover known as Multi-Step Crossover Fusion (MSXF). To see wheather we get good results by modifying the classical approach using GA, we have compared the result with the standard benchmark instances available. Using this combinations of the techniques, we have got results showing deviations less than 3%.

Keywords

Job Shop Scheduling, Genetic Algorithm, Giffler-Thompson Algorithm, Multi-Step Crossover Fusion (MSXF)

I. Introduction

The job shop scheduling problem is the combinatorial problem in which there can be many objective functions. In this paper we have selected the makespan to be the objective function. The main objective of any scheduling is to create a schedule which specifies when each task is to begin and what resources it will use that satisfies all the constraints while taking as little overall time as possible. The scheduling in which there is a specific order of a job to be processed by particular machines is called as job-shop scheduling problem. This kind of scheduling and sequencing of operations deals with the allocations of resources optimally over time so as to get optimum results that are maximizing the profit or minimizing the makesapn. To get the best desired result out of many different optimization techniques are used till now.

The job shop scheduling problem is considered a classic problem in the scheduling literature and it has received an enormous amount of attention. Scheduling can be seen as a problem to be searched for the optimal order and the times in which when a series of operations must be conducted under a particular set of conditions which must be satisfied so that the desired result is delivered on time. A planner is trying to use the means of production always as efficient and effective as possible and to minimize the time in which the entire process is handled which is also called as the makespan. However, it is very difficult to determine an optimal schedule. An exact method has very difficult to come up with the optimal solution in a reasonable time. But before some solution methods are presented, will be briefly discussing what is meant by job shop scheduling in this paper.

In the job shop scheduling problem there numbers of jobs J=(J1, J2, J3,…Jn) that are to be processed on the specified numbers of machines M=(M1, M2, M3,..Mm), where 'n' is the number of jobs and 'm' is the number of machines. The feasible solutions can be classified as semi-active, active, and non-delay schedules [1]. The set of non-delay schedules is a complete subset of the active schedules, where the active set itself is a complete subset of the semi-active schedules [1]. In the semi-active schedules, no operation can be locally shifted to the left, where in the active schedules, no left shift is possible either locally or globally. The two kinds of schedules may contain machine delay. Solutions having zero machine delay time are termed as the non-delay schedules. Job shop scheduling is the special type of scheduling where each job has a specific sequence on the predefined machines for an exact time interval individual of the other jobs.

Job-shop scheduling is usually done using heuristic algorithms [2] that take advantage of special properties of each specific instance. The concept and the need of the zero inventory has come up because of the current trends in market such as shorter product life cycles, variety in demands of the customers and the competitions of organizations to reduce the product cost and for this to happen the system within that particular organization must be very fast. This kind of contradictory requirement needs an efficient, effective and accurate scheduling techniques. Because of this reason only there is a need of good scheduling algorithm heuristic and meta-heuristics methods [3].In this paper, we have used Genetic Algorithm (GA) to solve this job shop scheduling problem. Genetic algorithms can be considered as one of the most popular techniques in evolutionary computation research, corresponding programming is called is genetic programming. Being a very complicated problem, job shop scheduling can be solved by a technique which basically an approximation technique since it has given best optimized results many times. This not only justifies the intensive research carried out in this field over the last several years but also the interest in finding new solutions to specific problem. Because sequencing problems are in general NP-hard, this means that there is no algorithm exists that can guarantee the optimum solution within the polynomial time. And is also is even one of the most difficult problems that can be solved in the group of NP-hard problems that have come across.

There are many other techniques have been invented in the field of Artificial Intelligence[4] and Operational Research over the years, where both general approach to problem-specific algorithms as exact solution techniques have been developed. Approximation Algorithms helps us to make sure that there is a solution, but this usually can come at the expense of the quality of the solution of the problem. Most of these algorithms are based on local decision rules. Hence, one can find the application of almost every artificial intelligence technique; for instance, logic programming[5], neural nets[6], machine learning, space state heuristic search, branch and bound, local search[7] and Genetic Algorithms (GAs), among others. Jain and Meeran[8] summarize the main techniques applied to solve a family of these problems, the job-shop scheduling problem, together with the category each technique belongs to. The application of GAs to scheduling problems has interested many researchers because they seem to offer the ability to cope with the huge search spaces involved in optimizing schedules. In this paper we will be focusing mainly on the classical, traditional job shop scheduling problem which is being solved by genetic algorithm. This is in fact well-known difficult to solve combinatorial optimization problem in which the Operational Research and Artificial Intelligence several other methods are worked out. Conventional solution methods can rarely (almost) optimal solution found within a reasonable time.

To solve this kind of stubborn problem many mathematical models or techniques called integer programming, mixed-integer programming [9], and dynamic programming. Also one method called branch and bound [10] is a enumerative technique. The basic idea of branch and bound is to conceptualize the problem as a decision tree. A node is basically a choice point of each decision and also it corresponds to a partial solution. From each node, there are new branches, one for each possible decision. This branching process continues upto the leaf nodes that cannot branch any further. To get the good computation speed for getting results, branch and bound technique becomes very expensive. A technique called Lagrangian relaxation [11], which is being used for more than around 30 years, does just this job nicely but somewhere around a point this method is also computationally expensive.

Dispatching rules[12] are applied to the problem of scheduling since long time. Use of these rules to the scheduling problem make it provide the good solution to complex problems in real-time. The dispatching rules are classified in many other types such as given below.

Table 1: Dispatching Rules

SOT(shortest operation time)	An operation with shortest processing time on the considered machine
2.LOT(longest operation time)	An operation with longest processing time on the machine considered
3.LRPT(longest remaining processing time)	An operation with longest remaining job processing time.
4.SRPT(shortest remaining processing time)	An operation with shortest remaining job processing time.
5.LORPT(longest operation remaining processing time)	An operation with highest sum of tail and operation processing time.
6. Random	The operation for the considered machine is randomly chosen.
7. FCFS (first come first served)	The first operation in the queue of jobs waiting for the same machine.
8.SPT(shortest processing time)	A job with smallest total processing time.
9.LPT(longest processing time)	A job with longest total processing time.
10.LOS(longest operation successor)	An operation with longest subsequent operation processing time.
11.SNRO(smallest number of remaining operations)	An operation with smallest number of subsequent job operations.
12.LNRO (largest number of remaining operations)	An operation with largest number of subsequent job operations.

In early 80's some new technologies were started and they were applied to job shop scheduling problem. Those methods or techniques collectively called as Artificial Intelligence (AI). Artificial techniques involves methods such as expert systems, knowledge-based systems. Latter, use both quantitative and qualitative knowledge in the decision-making process. First method, they are capable of generating heuristics that are significantly more complex than the simple dispatching rules described above. There can be serious disadvantages of this methods also. First of all they can be very time consuming as well as difficult to change and maintain. And since they generate feasible solutions, it can be very difficult to say how it is closed to the optimal solution.

Some neighborhood search techniques are very popular. When this method is combined with some other heuristic methods, it can be very effective as it gives a good solution. Some of the other methods are tabu search technique basic idea of which is to find all the feasible solutions by sequence of moves .A move from one schedule to another schedule is done by evaluating all candidate solutions and choosing the best one among available, just like a method called gradient-based technique. Tabu is nothing but a move which is put onto some tabu list which is built up from the history of moves used during some particular search. This tabu is moved for exploration of the search space until old solutions are left behind. This method also involves long term memory mechanisms [13]. Nowicki and Smutnicki (Glover 1996) implemented tabu search methods for job shop and flow shop scheduling problems. Ponnambalam et al. considered the tabu search technique for solving job shop scheduling problems with objective function used was makespan. The method used was adjacent pair wise interchange to generate the neighborhoods. The results of tabu search are compared with method of simulated annealing and genetic algorithms. It was cleared that the performance of tabu search method is comparable to that of genetic algorithm and simulated annealing. Vaessens (Glover 1996) showed that tabu search methods (in specific job shop scheduling cases) are superior over other approaches such as simulated annealing, genetic algorithms, and neural networks.

Method of simulated annealing [14]was being applied effectively to job shop scheduling problems. Vakharia and Chang (1990) developed a scheduling system based on this method for manufacturing cells. Jeffcoat and Bulfin (1993) applied the same to a resource-constrained scheduling problem [15]. Their results showed that this procedure provided the best results in comparison with other neighborhood search procedures. This method is an iterative search method.

II. Problem Formulation

In job shop scheduling problem, each job should be processed through the machines in a particular order only which is also known as technological constraint. Once any operation machine starts to process a job, no interruption is allowed till that job is over.

As we all know that job shop scheduling problem is considered as the most difficult combinatorial problem. We are using genetic algorithm to solve it. Thought it is an approximation method, it has got potential to give a desirable output. So to get the desired out i.e. the schedule which gives the minimum makespan, we have combined certain methods which are being to solve this problem. These methods are a Giffler-Thompson algorithm, a multistep crossover fusion along with neighborhood search techniques.

The job shop scheduling problem consists of n jobs and m set of machines, where each job j has to visit a number of machines in a predetermined order for completion. The processing times for each job at each machine are given and no machine can process more than one job at a time. Job-Shop Scheduling Problem (JSSP) can be defined as a problem is nothing but finding a job having a processing order at each machine that conforms to operations predetermined processing order of each job and minimizes or maximizes a given objective function. In this paper our functions makespan is to be minimized.

To simplify, we have made the following assumptions which, nevertheless, are not detrimental to the generality of the proposed method.

- No job can visit any machine more than once;
- Each job j has an operating sequence, there is a predefined set of processing times; for a given machine, and a given operation;
- An operation which has started runs to completion (non-preemption condition);
- Each machine can perform operations one after another(resource constraints);
- The operation times are fixed and predetermined (no setup-times allowed);
- Any operation time is represented as a positive integer number which specifies the duration of the operation in terms of time units;
- The time required to complete the whole job constitutes the makespan Cmax.

According to these statements, a Job shop schedule can be described by a matrix of n × m entries JS[n, m], where each entry consists of two integer numbers. The row JS[j] describes the list of the operations, according to their execution order, for the job j, while the entry JS[j, i] specifies the machine needed for the operation i and the corresponding processing time. Job shop problem can be represented in matrix form. Consider a job shop problem with n jobs that have to be processed on m machines.

There are several optimization objectives but the current paper will focus only on reducing the makespan (Cmax), which is measured in time units, for all our case studies and tests.

III. Methodology

The methodology used to find out the optimal result contains the combination of some other heuristic methods along with the classical genetic algorithm. There are three types of feasible job shop schedules.

1. Semi-active schedules: No operation can start earlier without altering the machine sequence.
2. Active schedules: No operation can start earlier without delaying another operation or without violating the precedence constraints.
3. Non-delay schedules: No machine is kept idle when there is an operation that can start on this machine.

A. Giffler-Thompson Genetic Algorithm

1. In the GT-GA, each individual is always a feasible schedule represented by a solution matrix. In fact, each individual is not only feasible but also an active schedule. We have no distinction between genotype and phenotype here.
2. Because the problem is a minimization problem, the rank-based roulette wheel selection [16] method is used.
3. An elitest strategy [17-18] to preserve the best individual in the current population to the next generation is used.

When the precedence constraints or other assumptions of the job shop scheduling problem is violated then the solution or schedule becomes infeasible. Following figure shows the relation between active schedules, semi-active schedules and non-delay schedules. It is very clear from the diagram that all the non-delay schedules are part of active schedules only, that means all the non-delay schedules are active schedules. And the optimal schedules are also active schedules. From the following figure we can say this classification of schedules clearly reduces the search space considerably.

B. A Giffler-Thompson Algorithm for Job Shop Scheduling [19]

As always, we are given a job shop scheduling problem represented by {Tjk}, the technological sequence matrix, and {p jk}, the processing time matrix. Besides, the following GA parameters are given: population size N, crossover rate Rc and mutation rate Rm.

A random initial population P(t = 0) of size N is constructed in which each individual is generated using the Giffler-Thompson(GT)[20] algorithm with randomly selecting operations. The makespan of each individual is automatically calculated A Genetic Algorithm based on the GT crossover is straightforward. The following points should be mentioned.

- Select randomly N × R cindividuals from P(t) and pair them randomly. Apply the GT crossover (with built-in mutation of probability Rm) to each pair and generate new N × Rc individuals that are inserted into P0(t). The rest of P(t) members are just copied to P0(t). As a result of the GT crossover, the makespan of each individual is automatically calculated.
- If the best makespan in P0(t) is not as good as that in P(t), then the worst individual in P0(t) is replaced by the best individual in P(t) (elitist strategy).
- Reproduce P(t + 1) from P0(t) by using the rank-based roulette wheel selection, in which each individual in P0(t) is sorted in the descending order of its makespan so that the worst individual is numbered as x1 and the best as xN. Then the roulette wheel selection is applied with the fitness f of an individual xi defined as f(xi) = i to obtain P(t + 1).
- Set t ← t + 1.
- Repeat from Step 2 to 5 until some termination condition is met.

Output the best individual in P(t) as the obtained best solution.

C. Multi-step Crossover Fusion (MSXF)

In a simple genetic algorithm, the role of the crossover operator is to function as a search engine, better strings can be constructed. But in genetic local search, local search plays the main role and the crossover, together with the particular selection operator, works as a navigation engine and helps find new starting points for the subsequent local search. Multi-step crossover (MSX)[21] is designed to be successful under such a genetic local search framework.

Although preliminary experiments in using JSSP benchmarks demonstrated the good performance of a GA with the MSX (GA/MSX), some computational disadvantages were found. Firstly the descent method which is used as a local search method in GA/MSX is too simplistic. Secondly, a lot of individuals are generated and evaluated during the MSX steps without contributing directly toward improving the solution quality. To reduce the computational time and improve the solution quality, MSX's functionality is incorporated into a neighborhood search algorithm and these two separate operators are fused together into a single unified operator called MSXF. a new crossover operator with a built-in local search functionality. The MSXF has the following characteristics compared to the neighborhood search crossover(NSX)[22]. It has following advantages.

- It can handle more generalized representations and neighborhood structures.
- It is based on a stochastic local search algorithm.
- Instead of restricting the neighborhood by a condition of intermediateness, a biased stochastic replacement is used.

D. Giffler-Thompson Genetic Algorithm with MSXF for JSSP

Initialize population: randomly generate a set of active schedules in equal number and apply the local search to each of them.

- Randomly select two schedules p0, p1 from the population with some bias depending on their makespan values.
- Change the direction (left or right) of p1 by reversing the job sequences with probability Pr.
- Do step (3a) with probability Pc, or otherwise do Step (3b). (a) If the DG distance between p1, p2 is shorter than some predefined small value, apply MSMF to p1 and generate q. Otherwise, apply MSXF to p1, p2 using the active CB neighborhood N(p1) and the DG distance and generate a new schedule q. (b) Apply Neighborhood search[23] with acceptance probability and the active CB neighborhood.
- If q's makespan is shorter than the worst in the population, and no one in the population has the same makespan as q, replace the worst individual with q.
  until some termination condition is satisfied.
- Output the best schedule in the population.

IV. Result and Analysis

On the basis of the results we have got by doing combination of techniques, comparison is shown as follows along with the percentage deviation. We have calculated the results with no. of generations constant at 700.

Sr. no.	Problem	Problem size	Best optimal result	Results we have got	% deviation	Computational time (sec)
1	mt 06	(6x6)	55	56	1.8182	8.324
2	Mt 10	(10x10)	930	933	0.3226	9.967
3	mt20	(20x5)	1165	1182	1.4592	14.123
4	la01	(10x5)	666	666	0	11.437
5	la02	(10x5)	655	655	0	11.693
6	la03	(10x5)	597	597	0	19.733
7	la04	(10x5)	590	601	1.8644	19.981
8	la05	(10x5)	593	593	0	9.953
9	la21	(15x10)	1046	1056	0.9560	36.782
10	la24	(15x10)	935	959	2.5668	35.897
11	la25	(15x10)	977	1008	3.1731	23.451
12	la27	(20x10)	1235	1278	3.4882	19.812
13	la29	(20x10)	1166	1212	3.9451	27.410
14	La 38	(15x15)	1196	1235	3.2609	30.167
15	La 40	(15x15)	1222	1264	3.437	32.908
16	orb01	(10x10)	1059	1071	1.133	21.065
17	orb02	(10x10)	888	903	1.6891	23.089
18	0rb03	(10x10)	1005	1028	2.2885	20.317
19	orb04	(10x10)	1005	1033	2.7861	19.389
20	orb05	(10x10)	887	896	1.0146	17.956
21	abz05	(10x10)	1234	1269	2.8363	20.447
22	abz06	(10x10)	943	969	2.7571	20.593

Along with these comparison with the standard benchmark instances we have made a graph of how the makespan is reducing gradually while number of generation is increasing. Following are the graph of some of the results.

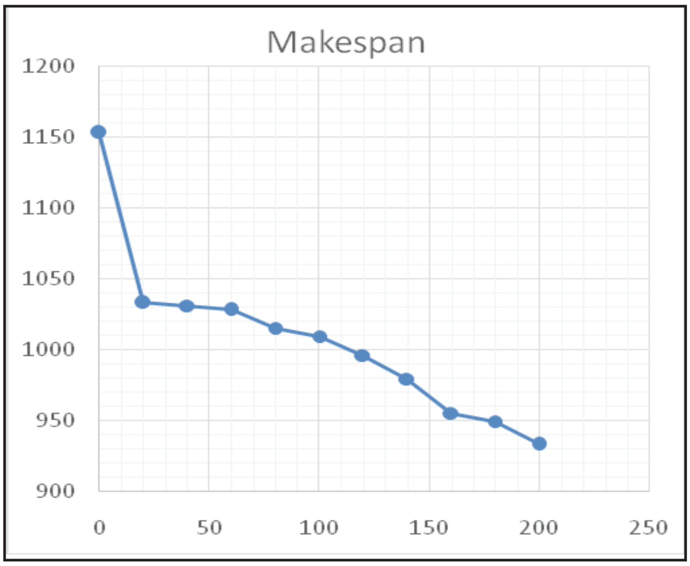

Fig. 1: Makespan Verses no. of Generations for mt10 Instance

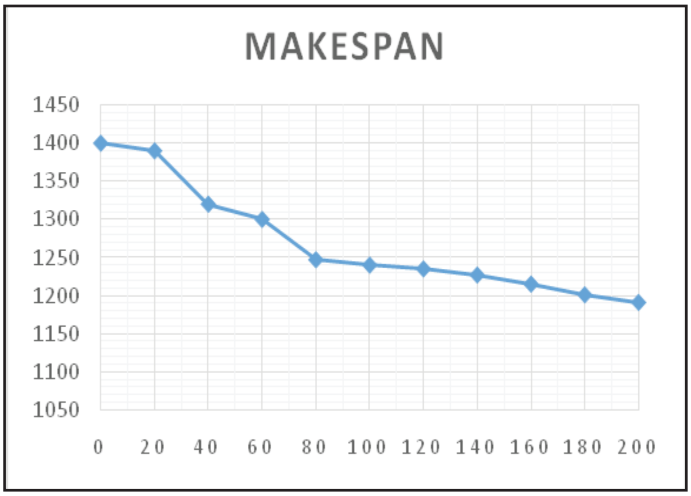

Fig. 2: Makespan Verses no. of Generations for mt20 Instance

We can see from both the graphs that as the no. of generations goes on increasing the makespan is decreasing gradually in both the cases towards best optimal value available. For the first graph the % deviation is just 0.3226 and for the second graph, curve shows how the makespan reduces gradually as no. of generations increases. So the technique or method which we have used can yield better results if we try it our for some more few thousands generations but since it will take more computational time it certain modifications are to be done further in future.

On the basis of the results what we have got we can say that some modifications in classical genetic algorithm along with some added techniques like Giffler-Thompson GA and use of MSXF along with local search can yield a good results. By introducing certain modifications or combination of different methods we can get even improve the quality of the result. As we can see from the comparison table, the results we have got for benchmark instances la01, la02, la03 having 0% of deviation that means we have got exact optimal result available. For benchmark instances like la05,mt10 and la21 are having less than 1% of deviation.

The mutation rate(probability) for our problem is 0.01% and crossover rate(probability) is 0.05%.

Computer used is having following configurations:

- CPU info : Intel i3 CPU @ 2.27Ghz
- CPU Cores : 2
- RAM : 4GB

So we can conclude that GT-GA algorithm with some additional features of local search, neighborhood search we can get quality of results. For further research, the techniques which are used in this paper can be applied to a larger size problem to see how it is performed.

References

[1] José Fernando Gonçalves, Jorge José de Magalhães Mendes, Maurício G.C. Resende,"A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem", AT&T Labs Research Technical Report TD-5EAL6J, September 2002.

[2] Kamaljit Kaur, Amit Chhabra, Gurvinder Singh, "Heuristics Based Genetic Algorithm for Scheduling Static Tasks in Homogeneous Parallel System", In International Journal of Computer Science and Security (IJCSS), Vol. 4, Issue 2.

[3] San Nah Sze, Wei King Tiong,"A Comparison between Heuristic and Meta-Heuristic Methods for Solving the Multiple Traveling Salesman Problem", World Academy of Science, Engineering and Technology International Journal of Mechanical, Aerospace. Industrial and Mechatronics engineering Vol. 1, No. 1, 2007.

[4] M. Vidyasagar, Vincent D. Blondel,"Probabilistic solutions to some NP-hard matrix problems", In Automatica 37, 2001, pp. 1397-1405, Elsevier.

[5] Philippe Laborie,"Algorithms for propagating resource constraints in AI planning and scheduling: Existing approaches and new results", Artificial Intelligence 143, 2003, pp. 151–188. Elsevier.

[6] M. Anton Ertl, Andreas Krall,"Optimal Instruction Scheduling using Constraint Logic Programming", In Programming Language Implementation and Logic Programming (PLILP) 1991, Springer LNCS 528, pp. 75-86.

[7] A. S. Jain, S. Meeran,"Job-shop scheduling using neural networks", International Journal of Production Research, May 1998, 36(5), pp. 1249-1272.

[8] Jean-Paul Watson, J. Christopher Beck,"A Hybrid Constraint Programming / Local Search Approach to the Job-Shop Scheduling Problem", In L. Perron and M. Trick (Eds.): CPAIOR 2008, LNCS 5015, pp. 263–277, 2008, Springer-Verlag Berlin Heidelberg 2008.

[9] A.S. Jain, S. Meeran,"Deterministic Job-shop Scheduling: Past, Present and Future", European journal of operational research, 113(1999), pp. 390-434, Elsevier.

[10] MilošŠeda,"Mathematical Models of Flow Shop and Job Shop Scheduling Problems", International Journal of Applied Mathematics and Computer Sciences, Vol. 4, No. 4.

[11] Peter Brucker, Bernd Jurisch, Bernd Sievers,"A branch and bound algorithm for the job-shop scheduling Problem", Discrete Applied Mathematics 49, 1994, pp. 107-127, Elsevier.

[12] HaoxunChen, Peter B. Luh,"An alternative framework to Lagrangian relaxation approach for job shop, European Journal of Operational Research 149, 2003, pp. 499–512, Science direct, Elsevier.

[12] AtifShahzad, Nasser Mebarki,"An alternative framework to Lagrangian relaxation approach for job shop Scheduling", 8thInternational Conference of Modeling and Simulation - MOSIM'10 - May 10-12, 2010 – Hammamet– Tunisia, "Evaluation and optimization of innovative production systems of goods and services".

[14] Jean-Paul Watson,"On Metaheuristic "Failure Modes": A Case Study in Tabu Search for Job-Shop Scheduling", In MIC2005: The Sixth Metaheuristics International Conference, Vienna, Austria, August 22– 26, 2005.

[15] Peng Tian, Jian Ma ,Dong-Mo Zhang,"Application of the simulated annealing algorithm to the combinatorialoptimisation problem with permutation property:An investigation of generation mechanism", In European Journal of Operational Research 118, 1999, pp. 81-94, Elsevier.

[16] K. Bouleimen, H. Lecocq,"A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version", European Journal of Operational Research 149, 2003, pp. 268–281, Elsevier.

[17] Omar Al Jadaan, Lakishmi Rajamani, C. R. Rao, "Improved selection operator for ga", In Journal of Theoretical and Applied Information Technology, pp. 269-277.

[18] Zhihua Tang, Youtuan Zhu, Guo Wei, Jinkang Zhu, "An Elitist Selection Adaptive Genetic Algorithm for Resource Allocation in Multiuser Packet-based OFDM Systems", in Journal of communications, Vol.3, No. 3, july 2008.

[19] Shengxiang Yang,"Genetic Algorithms with Memory- and Elitism Based Immigrants in Dynamic Environments", In Evolutionary Computation 16(3): By the Massachusetts Institute of Technology, pp. 385–416, 2008

[20] Takeshi Yamada, Ryohei Nakano,"Job-shop scheduling", in IEE control engineering series 55, The Institution of Electrical Engineers.

[21] Mark Moone, Gerrit K. Janssens,"A Giffler-Thompson Focused Genetic Algorithm for theStatic Job-Shop Scheduling Problem", Centre for Industrial Management, Katholieke Universiteit Leuven, Celestijnenlaan 300A, B-3001, Heverlee-Leuven, Belgium.

[22] Takeshi Yamada, Ryohei Nakano,"A Fusion of Crossover and Local Search", IEEE International Conference on Industrial Technology (ICIT '96) Shanghai, China December 2 - 6, 1996 pp. 426-430.

[23] Mehmet Sevkli, M. EminAydin,"Variable Neighbourhood Search for Job Shop Scheduling Problems", Journal of software, Vol. 1, No. 2, August 2006.

Harmesh Kishor Vanmali received his B.E. degree in Mechanical Engineering from 'Vidyavardhini's College of Engineering', Mumbai University, India, in 2011. He is pursuing his M.E. (Master's degree) in Mechanical Engineering (CAD-CAM with Robotics). He is currently working as lecturer in 'Shree L. R. Tiwari College of Engineering', Mira Road (E), Mumbai, India, from July 2012. His research interests include optimization techniques in operational research.

ISSN : 2249-5762 (Online) | ISSN : 2249-5770 (Print)

Prof. V. S. Jorapur received his B.E. degree in Mechanical Engineering from Karnatak University Dharwar, India, in 1986, the M.E. degree in Production Technology from Karnatak University, Dharwar, India, in 2005, and pursuing Ph.D. degree in job shop scheduling and genetic algorithms from Visveswaraya Technolgical University Belgaum, India. He is currently working as Associate Professor, with Department of Production Engineering, FR. Conceicao Rodrigues College of engineering Mumbai, India.