Application of GA and PSO for Flow-Shop Scheduling Problem

[1]Deepanshu Arora,[2]Gopal Agarwal
[1]Research Scholar,[2]Professor
[1]Department of Mechanical Engineering,
[1]Malviya National Institute of Technology, Jaipur,India

***Abstract*—In this paper, we present GA and PSO for multi-objective function for flow-shop scheduling problem with various weighted sets. For this work we used the multi-target function which includes four objectives are total weighted squared tardy, total weighted squared earliness, maximum Makespan, no. of tardy jobs. The multi-objective function contained here is the minimization of weighted sum of all four objectives. As increases the size of the problem solving time also increases (NP-Hard Problem) so we use the Meta heuristic approach to solve the problem. Genetic Algorithm GA and Particle Swarm Optimization PSO are the most important technique of meta-heuristic approaches capable to handle Nonlinear NP-Hard large size of problems. Comparison between the results of the GA and PSO shows that in larger problems the results obtained by PSO do not have adequate efficiency compare to GA.**

***Index Terms*—Flow shop scheduling problem; genetic algorithm(GA); particle swarm optimization (PSO)**

I. INTRODUCTION

Broadly speaking the schedule is a document, which provides information about encountering things, how they are going to be happen and manage by providing plan based on start and end time of certain activities. Scheduling is the method of creating sequence and schedule for manufacturing systems. When the need of scheduling arises, it can be managed by the proper sequencing in which planning is for decisions.These decisions are for selecting the next task and the starting and completing time of each task.

Flow-shop Scheduling Problem (FSSP) is one of extremely hard problems because it requires very large combinatorial search space and the precedence constraints between machines. FSSP is a difficult NP-hard combinatorial optimization problem. The traditional algorithm used to solve the problem is the branch-and- bound method, which takes considerable computing time when the size of problem is large. And also itprovides only single solution. The study carried out and the results presented are based on Meta-heuristic approaches for solving FSSP. Meta-heuristic approaches generate large set of possible solutions.

II. CLASSIFICATION OF OPTIMIZATION TECHNIQUES

The optimization techniques for scheduling problems are classified in two types, one is continuous and the second is combinational. Most of the continuous kind of optimization techniques having traditional kind of methods like linear programing, quadratic programing, and nonlinear kind of branch and bound methods. But the continuous optimization techniques are the new and global kind of optimization techniques those having ability to work for a large number of data and perform mathematical & computational formulation, so that they are easy to handle very hard nonlinear kind of scheduling problems. In this work we focused on the NP-hard kind of Flow Shop Scheduling problem and having a review of past researchers those have already applied Genetic Algorithm to solve it. In this paper we are presenting the way, how the past researchers used GA and PSO for FSSP and a comparative analysis of performance between GA and PSO. [6]

III. MATHEMATICAL MODEL

Multi-objective function FSSP: we follow the multi-objective fitness function for Flow shop scheduling which is based on the due dates, processing time and setup time according sequence, where constraints of objective functions are 'Total Squared Tardiness', 'Makespan', 'Total Squared Earliness' and 'Total numbers of Tardy Jobs'. The multi-objective function is to minimize all the objectives simultaneously for flow shop scheduling. Because the target of research is to minimize the overall Makespan of the flow shop scheduling by which we can increase the productivity, and used the resources with effective and efficient manner. As we know, time is directly related to cost and this multi-objective function will provide a unique solution considering all relevant constraints simultaneously. [1, 2].Deb, K [5] explain the application of evolutionary approaches for multi objective problems in his book. Azardoost E. B. [8] used hybrid algorithms for flow shop scheduling with two objectives.

The procedure for optimizing the FSSP is to minimize Makespan first, so it will ultimately meet due dates efficiently.If the machines are used very effectively then only it can be possible, this is the main resign for the requirement of optimum scheduling to minimize Makespan. Here we followed the multi-objective fitness function based on the work by AshwaniDhingra[7] for driving the optimum flow shop scheduling results. By use of this optimum schedule,Makespan has also optimized. The objective function has four constraints i.e. Total Weighted Squared Tardiness, Makespan, Total Weighted Squared Earliness and Total numbers of Tardy Jobs [7]. It is good to handle all constraints simultaneously; because these constraints directly affect to meet the due dates. The multi-objective function and its relevant constraints are as follows.

Multi-objective function:

The multi-objective function and its constraints are as below:

$$f(x)_{min} = \left(\sum_{j=1}^{n} w_j T_j^2 + C_{max} + \sum_{j=1}^{n} w_j E_j^2 + (N_t) \right)$$

1. *Total Weighted Squared Tardiness-*

This measure reflects the due dates of jobs to be reschedule for minimization of late deliverables and is defined as follows.

$$\sum_{j=1}^{n} w_j T_j^2$$

$$T_j = Max\{C_j - d_j\} \text{ if } \quad C_j - d_j \geq 0$$

$$= 0 \textit{ otherwise.}$$

Here T_j = Tardiness,

C_j = Completion time,

$[C_j - d_j = L_j]$ and L_j = lateness

2. *Makespan-*

$$C_{max} = Max(C_1 ....C_n)$$

3. *Total Weighted Squared Earliness*

This measure reflects the due dates of jobs to be rescheduled for minimization of early deliverables and is defined as follows.

$$\sum_{j=1}^{n} w_j E_j^2$$

$$E_j = Max\{d_j - C_j\} \textit{ if } \quad d_j - C_j \geq 0$$

$$= 0 \textit{ otherwise.}$$

4. *Total numbers of Tardy Jobs*

This measure reflects the total number of jobs considered for minimization.

$$N_t = \sum_{j=1}^{n} U_j$$

IV. META-HEURISTIC ALGORITHMS FOR FSSP

A.Genetic Algorithm (GA):

Sergio Cavalieri and P. Gaiardell[22] used the Genetic Algorithm GA by two ways. First was based on allocation and second was on sequence production lotsin job shop. There the problem was based on nonlinear, multi-objective fitness function. L. D. Ravindran& A. NoorulHaq[4] used GA approach to optimize the value of Makespan and minimize it, and also for the total flow time for job shop scheduling. He used three hybrid algorithms based over genetic algorithm. T .Pasupathy, ChandrasekharanRajendran& R.K. Suresh [24] followed the flow shop scheduling problem, where the scheduling problem was considered with bi-objective of minimizing Makespan and total flow time of jobs based over multi-objective genetic algorithm. Chuen-LungChenet. al. [3] also used GA for flow shop scheduling problem. The work was based on Taillard[26] where numbers of jobs were varying from 20 to 200 and numbers of machines were varying from 5 to 20.

B.Particle Swarm Optimization (PSO):

Ruiz and Stützle [20]used particle swarm optimization for permutation flow shop scheduling. They followed the total flow time as single objective. Their research was based on competitive analysis of PSO with discrete differential evolutionary approaches. T.C. Wong, Felix T.S. Chan and L.Y. Chan [27]used PSO with lot and sub lots, the sub-lot having same size and provide better results Li. D. and Deng, N [12]proposed an electoral cooperative PSO based on several sub swarm for permutation FSSP, by individual voting for sub swarm. They applied this approach to benchmark Taillard's [26] and the results were good performance of electoral cooperative PSO for permutation FSSP.

V. EXPERIMENT INPUTS

In this work we used computational experiments to solve the multi-objective optimization problem for comparative analysis between GA and PSO for their performance. In this experimental work for optimising flow shop scheduling on the basis of best fitness value for problem size of 5, 10 and 15 machines with 10, 20, 25, 50 and 100 jobs. Four input variables used here are processing time for individual job on individual machine, due date for each job, weighted value for each job and sequence dependent setup time for each machines.The same weighted value for every job is taking as assumption in this experiment setting.

VI. EXPERIMENTAL SETUP AND RUN

Genetic algorithm is the one of the best method of evolutionary algorithms, for multi-objective optimization problem specially. And here we perform experiments to solution sequence dependent FSSP through GA and PSO on MATLAB platform. So here we use Genetic Algorithm as a universal search method of finding the optimum solution, same for the PSO also. The multi-objective fitness function of the problem has constraints which are already discussed.

A. Setup for GA and Its operators:

Now Genetic Algorithms use for obtain a basic seed sequence based on the multi objective fitness function for generating initial population This basic seed sequence along with set of (Population Size - 1) random generated initial population size.And by selecting a good set of chromosomes caring minimum fitness value in the initial population usually improve the performance of genetic algorithm. GA start the optimization process with seed sequence, then GA generate random population, then it combined with initial seed sequences. The GA operators perform creates a sequence of new population. With every stage GA uses the fitness evaluation in current generation for selecting best chromosomes for next population. The fitness value is based on the multi-objective function, which we are taken for optimising FSSP followed by Total weighted Squared Tardiness, Makespan, Total weighted Squared Earliness and Total numbers of Tardy Jobs simultaneously. After that algorithm select the seed chromosomes, and according to fitness value call the parental generation. Parents are selected according their fitness value (as selection criteria), according to our objective function which is minimization function; minimum fitness value is choose as base fitness value. The individual members of the current population that have lesser fitness value are chosen as Elite. These Elite are member of current population which is considered to the next population. It produces off-springs from the parent. Those offspring are child or new generation. There are two methods of generating off-springs first, by Crossover and second, by Mutation. By crossover off-springs are produced either by combining the vector entries of pair of parents, and by mutation only random changes is done on single parent. And then replace the current generation with their children for next generation. And this process is run in continuously till optimum solution is obtained. The end run of the algorithm is computational time based. Algorithm run continuously and does not stop until time limit reaches $n \times m \times 0.5$ seconds. Where n refers to the number of jobs and m is the numbers of machines in the system. Generation of seed sequence in first step has been obtained from genetic algorithms. [1, 2]

Where, n= no. of jobsand m= no. of machines.In this case we followed following settings

Population size for each generations	= 50
Elite Count	= 02
Crossover (Order) probability	= 0.8
Mutation (Reciprocal Exchange) probability	= 0.1 to 0.15
Selection of chromosome Strategy	= Roulette wheel
Stopping condition of iteration loop	= (n × m × 0.5) Sec.
Migration rate in one generation	= 0.2
Migration Interval	= 20

Genetic Algorithm Parameters:

i. Population Size:

Population refers to the set of solutions or results.The population is used to show the particular generation and first population contains large number of diversified solutions but after the generation of new population, we have better results. In computational sector we fix the population size less than the available number of sequences so that it will take lesser time and provide better results. In present work we fix the size of population as 50.

ii. Selection:

Select those chromosomes with promising fitness value and participate for new generation with better results.The selection procedure is based on the individual fitness value of chromosome.These best chromosomes show the best quality for the available multi-objective optimization problem.Best fitness value's chromosomes are recombined. Now they are ready to participation for new generation of better results.Here in present work we followed the straight forward selection procedure, Roulette wheel selection for selection of best chromosomes.

iii. Crossover function:

Crossover is used for generating new set of solution (new generation).The characters of chromosomes carried by genes are mutually interchanged between two chromosomes.New types of child chromosomes develop with the same characters like parental chromosomes and with better also.There are so many types of crossover possible like partial matched crossover, order crossover, cycle crossover and single point crossover etc.In present work the rate of crossover is fixed to 80 per cent, which is a good crossover rate for generating better results.

iv. Mutation function:

It is the last operator of Genetic Algorithm, which is also used for generating new generation.Mutation generation of single child is done by single parent by a small random Reciprocal exchange or inversion of individual characterfor a solution. The probability of mutation function is lesser then the crossover, individually it is very small. In our work the mutation rate is also fixed with 10 to 15percent.

v. Elite Count:

After the generation of new population, next step is to decide which individuals to keep and which ones to discard.

By the generational replacement the new generation supersedes the old generation, and there is always a risk of dismissing of very promising solutions.So here the most important Elitism concept is developed, which is used to always keep the most promising solution in the population.

vi. End-run condition:

End-run condition is the condition used in genetic algorithm for stopping the process of generation or running off algorithm computationally. We use end-run limit as $n \times m \times 0.5$ seconds.

B. Setup for PSO and Its operators:

PSO is a neighborhood optimization algorithm that models the social behavior of particles within a search space. PSO was first introduced by Kennedy and Eberhart [11]. They introduced PSO as a simulation of behavior of bird for search optimal solution, but it quickly evolved into one of the most powerful optimization algorithms in the computational intelligence field. The algorithm's principle has partials in n-dimensional search space. The position of each and every particle represents a potential solution to the optimization problem. This position is used to determine the fitness value of a particle. In each next iteration, partials update their location on the bases of two things: one is the best location (or solution) found by itself, the*pbest*. The second is the one found by the whole swarm by distance, the*gbest*. Every particle updates its velocity and position according to the following equations.

$$V_{(i,j)}^{t+1} = w.V_{(i,j)}^{t} + c_1 r_1 \left\{ \left(pbest_{(i,j)}^{t} - X_{(i,j)}^{t} \right) \right\} + c_2 r_2 \left\{ \left(gbest_{(i,j)}^{t} - X_{(i,j)}^{t} \right) \right\} \quad \text{(i)}$$

$$X_{(i,j)}^{t+1} = X_{(i,j)}^{t} + V_{(i,j)}^{t+1} \quad \text{(ii)}$$

Where, equation (i) is used for velocity update of each particle between two consecutive points, here first point is the past location of particle and the second point is the current location of that particular particle. And the equation (ii) is used to locate particles in solution space; here it used the past location of a particular particle and its current velocity which it used to move from past location to current location. Here n= no. of jobsand m= no. of machines. i= no. of particlesand j= no. of dimensions.In this case we followed following settings

No. of dimension = 04
Numbers of particles = 50
Numbers of iteration = 500
C_1 and C_2 = 2.05
Initial velocity = rand (50, 4) *(Randomly taken)*
Initial positions = rand (50, 4) *(Randomly taken)*
Stopping condition of iteration loop = (n × m × 0.5) Sec

Sequence generator generate unique sequence for every partial, total number of sequences are equal to !n, number. of jobs. Objective function solver was based on integer programing for each and every sequence of jobs; it calculates the final optimal value in terms of fitness value. Selection of *gbest* and *pbest* were based on fitness value for each and every string. Then according the principle of PSO compare the current fitness value of each string with past string's fitness value and follow the best and again select *gbest* among all possible solutions for that particular iteration.

The algorithm carried out until all the iteration has been completed or provide a time limitation of$n \times m \times 0.5$ seconds or till the system stabilizes.

VII. Results and Analysis

Experimental observations are the values as the output of the MATLAB. Here we developed MATLAB codes for solving FSSP by GA and PSO. The output of the experiment is taken as fitness value. Both GA and PSO approaches used for same size of problem and theperformance measure are taken in form of relative percentage deviations (RPD). The relative percentage deviation is the most common performance measure to compare two or more algorithms [17]. Here the benchmark has been taken as the Taillard's [26] problems in this experiment.

Table 1 Table Type Styles

RELATIVE PERCENTAGE DEVIATION FOR GA AND PSO			
S. No.	Problem Sizes	GA-RPD	PSO-RPD
1	5X10	0.00%	0.08%
2	5X20	0.22%	0.84%
3	5X25	1.76%	4.51%
4	5X50	4.91%	6.83%
5	5X100	2.34%	3.76%
6	10X10	0.46%	0.91%
7	10X20	1.30%	3.55%
8	10X25	0.99%	2.79%
9	10X50	5.69%	6.51%
10	10X100	2.71%	4.45%
11	15X20	3.92%	6.98%
12	15X25	4.44%	4.68%
13	15X50	6.75%	3.27%
14	15X100	4.02%	4.11%

The above results were obtained by solving the Flow Shop scheduling problem in MATLAB software. Out of many sets of results we choose one set of solution depending upon the best fitness value. The current problem is for minimization of the objective function thus we consider the minimum fitness value. The following set of solution includes the optimum value for all the constraints related to objective function. From the resultant processing time obtained from the calculation we can clearly spot optimum Makespan value. In addition to the above, we also get the optimum schedule for the jobs.

Comparative results between GA & PSO are based on Relative Percentage Deviation (RPD) are in depicted in Figures 1-3: As shown in Figure 1, 5 machines problems with different number of jobs, the performance measure for the purpose of comparing GA and PSO approach is followed by RPD. However, for 5 machines problem initially both the GA and PSO perform almost same, but as the number of jobs increased we can see the value of RPD for GA is 0.20% and for PSO is 0.84%. That means GA performs comparatively better than PSO at this stage. Similarly with increasing size of jobs and with same number of machines as 5, we observed that the GA having 3.73%, 4.91% and 3.24% values for RPD comparatively PSO having 5.51%, 8.83%, 4.53%. It shows that with the large size of flow shop scheduling problems GA provides better results compare to PSO. Even if we increase the number of machines with large numbers of jobs, GA having much more attractive results then PSO, because GA having lesser value of relative percentage deviation. And with the lesser value of RPD shows more promising optimized results.

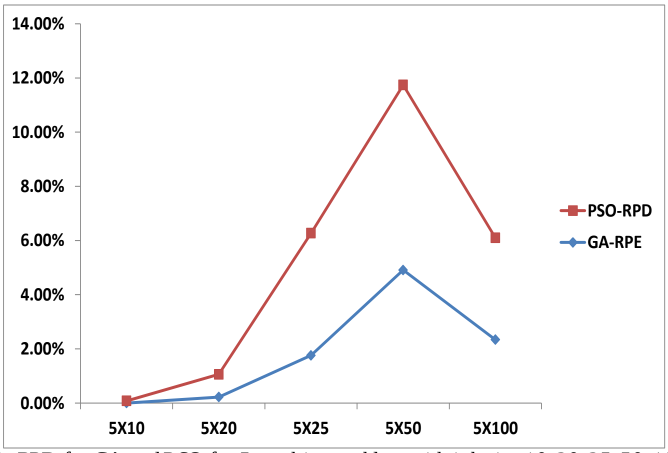

Fig.1. RPD for GA and PSO for 5 machine problem with job size 10, 20, 25, 50, 100

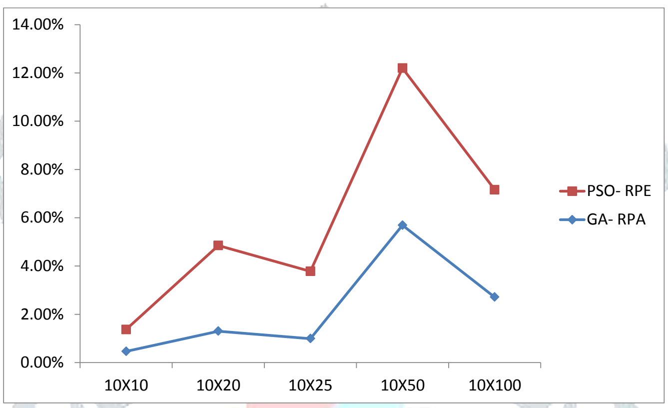

Fig.2. RPD for GA and PSO for 10 machine problem with job size 10, 20, 25, 50, 100.

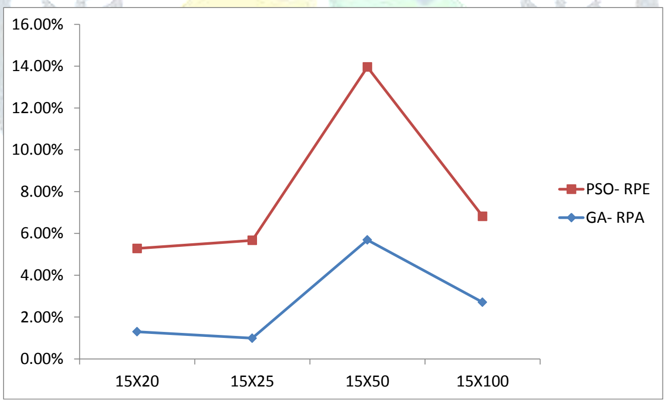

Fig.3. RPD for GA and PSO for 15 machine problem with job size 20, 25, 50, 100

VIII. CONCLUSION

Here an efficient GA and PSO algorithm has been presented to solve sequence dependent setup time based flow shop scheduling problem. Both the approaches minimize the total weighted squared tardiness, Makespan, total weighted squared earliness and number of tardy jobs. Due to the nonlinearity and the large-size of problem, we followed PSO to solve the present problem. The performances and results were compared with the GA solutions. PSO answers have a large variation in terms of Relative Percentage deviation. Also the PSO results are lesser attractive compare to GA results. The results obtain from both techniques got optimal solutions but the GA results are much more near to optimum solution.

REFERENCES

[1] AshwaniDhingra, PankajChandna, "Hybrid Genetic Algorithm for Multicriteria Scheduling with Sequence Dependent Set up Time". International Journal of Engineering, vol. 3, no. 5, pp. 510-520, 2009.

[2] AshwaniDhingra, PankajChandna. "A bi-criteria M-machine SDST flow shop scheduling using modified heuristic genetic algorithm". International Journal of Engineering, Science and Technology, vol. 2, no. 5, pp. 216-225, 2010.

[3] Chuen-Lung Chen, Venkateswara S. Vempati, and Nasser Aljaber., “An application of genetic algorithms for flow shop problems”. European Journal of Operational Research, no. 80, pp. 389-396, 1995.

[4] D. Ravindran, A. NoorulHaq, S.J. Selvakuar, and R. Sivaraman. “Flow shop scheduling with multiple objective of minimizing Makespan and total flow time.” International Journal of Advanced Manufacturing Technology, no. 25, 1007-1012, 2005.

[5] Deb, K., “Multi-Objective Optimization Using Evolutionary Algorithms”. WILEY-INDIA, 2002, Ch.1.

[6] DeepanshuArora, and GopalAgarwal., “Meta-heuristic approaches for flowshop scheduling problems: a review”. International Journal of Advanced Operations Management vol. 8, no. 1, pp. 1-16, 2016.

[7] AshwaniDhingra. “Multi-Objective Flow Shop Scheduling Using Metaheuristics”, National Institute of Technology Kurukshetra, 2010.

[8] ElnazBaghalAzardoost and NrgesImanipour, “A Hybrid Algorithm for Multi Objective Flexible Job Shop Scheduling Problem” International Conference on Industrial Engineering and Operations Management, pp. 22-24. Kuala Lumpur, Malaysia 2011.

[9] FatmaTangour, and IhsenSaad. “Multiobjective Optimization Scheduling Problems by Pareto-optimality in Agro-alimentary Workshop”, International Journal of Computers, Communications & Control , vol. 1, no. 3, pp. 71-83, 2006.

[10] GoldBerg, D. E., “Genetic Algorithm in search, Optimization and Machine Learning”. PEARSON 2005, ch.8.

[11] JocCingTay and NhuBinh Ho, “Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems”. Elsevier, Computers & Industrial Engineering, no. 54, pp. 453-473, 2008..

[12] Desheng Li, and Na Deng b., “Solving permutation flow shop scheduling problem with a cooperative multi-swarm PSO algorithm”, Journal of Information & Computational Science , vol. 9, no. 4, pp.977–987, 2012.,

[13] Ahmed El-Bouria, SubramaniamBalakrishnanb, and Neil Popplewellb, “A neural network model and algorithm for the hybrid flow shop scheduling problem in a dynamic environment. Springer”, Journal of Intelligent Manufacturing, no. 16, pp. 361-370, 2005.

[14] Masaya Yoshikawa, Hideto Nishimura, and HidekazuTerai, “A New Genetic Coding for Job Shop Scheduling Problem Considering Geno Type and Pheno Type”, Recent Advances in Computer Engineering and Applications, pp. 59-62.

[15] Mohammad Akhshabi, MostafaAkhshabi, and JavadKhalatbar, “Parallel Genetic Algorithm to Solving Job Shop Scheduling Problem”, Journal of Basic and Applied Scientific Research vol. 1, no.10, pp. 1484-1489, 2011.

[16] B. Naderi, M. Zandieh, A. KhaleghiGhosheBalagh, and V. Roshanaei, “An improved simulated annealing for hybrid flow shops with sequence-dependent setup and transportation times to minimize total completion time and total tardiness”, Expert Systems with Applications: an International Journal, no. 36, pp. 9625-9633, 2009 .

[17] N.S. Hemant Kumar and G. Srinivasan, “A genetic algorithm for job shop scheduling - A case study”, Elsevier, Computers in Industry no. 31, pp. 155-160, 1996.

[18] R. Ramezanian, M.B. Aryanezhad and M. Heydari, “A Mathematical Programming Model for Flow Shop Scheduling Problems for Considering Just in Time”, International Journal of Industrial Engineering & Production Research, vol. 21, no. 2, pp. 97-104, 2009.

[19] RuiZhanga, ShijiSongb, and Cheng Wub, “A hybrid differential evolution algorithm for job shop scheduling problems with expected total tardiness criterion” Elsevier, Applied Soft Computing, vol. 13, pp. 1448–1458, 2013.

[20] Ruiz, R. and Stützle, T., “A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem”, European Journal of Operational Research, vol. 177, no. 3, pp.2033–2049, 2007.

[21] Seetharama L. and Narasimhan, D. W., “Production Planning and Inventory Control”, New Delhi, Pearson, 2010, ch.10.

[22] Sergio Cavalieri and Paolo Gaiardelli, “Hybrid genetic algorithms for a multiple-objective scheduling problem”, Journal of Intelligent Manufacturing, vol. 9, pp. 361 – 367, 1998.

[23] VelusamySubramaniam, “Scheduling of Manufacturing System based on Extreme Value Theory and Genetic Algorithms”, Massachusetts Institute of Technology, 1995.

[24] T. Pasupathy, ChandrasekharanRajendran, and R.K. Suresh, “A multi-objective genetic algorithm for scheduling in flow shops to minimize the Makespan and total flow time of jobs”, International Journal of Advanced Manufacturing Technology vol. 27, pp.804-815, 2006.

[25] Taillard, E., “Some efficient heuristic methods for the flow shop sequencing problem”, European Journal of Operational Research vol. 47, pp. 65-74, 1990.

[26] Taillard, E., “Benchmarks for basic scheduling problems”, European Journal of Operational Research vol. 64, pp. 278-285, 1993.

[27] T.C. Wonga, Felix T.S. Chan, and L.Y. Chan, “A resource-constrained assembly job shop scheduling problem with lot streaming technique”, Computers & Industrial Engineering, vol. 57, no. 3, pp.983–995, 2009.

[28] Yi Zhang, Xiaoping Li, and Qian Wang, “Hybrid genetic algorithm for permutation flowshop scheduling problems with total flowtime minimization”. European Journal of Operational Research vol. 196, pp. 869-876, 2009.

[29] Z.X. Guo, W.K. Wong, S.Y.S. Leung, J.T. Fan, and S.F. Chan “Mathematical model and genetic optimization for the job shop scheduling problem in a mixed- and multi-product assembly environment: A case study based on the apparel industry”, Computers & Industrial Engineering, vol. 50, pp. 202-219, 2006.

[30] Zalinda Othman, KhairanumSubari and NorhashimahMorad, "Job Shop Scheduling With Alternative Machines Using Genetic Algorithms". JurnalTeknologi , 41(D), pp. 67-78, 2004.

[31] L. Wang and D.-Z. Zheng, "An Effective Hybrid Heuristic for Flow Shop Scheduling", International Journal of Advanced Manufacturing Technology vol. 21, pp. 38-44, 2003.