Original Article

Bulletin of Pure and Applied Sciences.
Vol. 36E (Math & Stat.), No.2, 2017.P.101-111
Print version ISSN 0970 6577
Online version ISSN 2320 3226
DOI: 10.5958/2320-3226.2017.00013.3

A HEURISTIC GENETIC ALGORITHM FOR BI-CRITERIA FLOW SHOP SCHEDULING WITH FUZZY PROCESSING TIME AND DUE TIME

Harendra Kumar*, Pankaj Kumar* and Manisha Sharma**

Author Affiliation:
* Department of Mathematics and Statistics, Gurukula Kangri University, Hardwar 249404, Uttarakhand, India
** Department of Mathematics, Panjab University, Chandigarh 160014, Punjab, India

Corresponding Author:
Pankaj Kumar, Department of Mathematics and Statistics, Gurukula Kangri University, Hardwar 249404, Uttarakhand, India

E-mail: pmittalvce@gmail.com

Received on 13.06.2017, **Accepted on** 20.09.2017

Abstract

The bi-criteria flow shop scheduling problem is widely studied optimization problem in which every machine has same work function and a job can be processed by any of available machines. The present research work includes a new genetic algorithm for solving the bi-criteria flow shop scheduling problem. The proposed genetic algorithm has own coding, evaluation function, crossover and mutation to minimize the maximum tardiness and weighted flow time. In this paper, cycle crossover operators for crossover and interchanging mutation operators for mutation are used. The objective of this paper is to find an optimal scheduling of 'n' jobs for 3 machines involving processing times, due time and weightage of jobs. The processing times of jobs have been considered as a fuzzy number to denote uncertainty of processing time which is more realistic and general in nature. The fuzzy processing times are defuzzified and converted into crisp one using fuzzy number ranking method. The present algorithm is collated with beforehand released problems in literature. The proposed algorithm is formulated and applied to numerical examples to demonstrate its effectiveness.

Keywords: Scheduling, genetic algorithm, fuzzy processing time, maximum tardiness, weighted flow time, due time.

1. INTRODUCTION

Scheduling is a subsidiary function of the production planning phase which is responsible for combining information on market demands, production capacity and current inventory levels to determine aggregate production levels in medium to long range horizons. The complexity of the scheduling problem increases with the increase in restrictions and conditions added to the problem. One aspect that increases the complexity of the problem is the consideration of more than one objective to optimize. In flow shop scheduling system, jobs flow from an initial machine through numerous intermediate machines and at last to a final machine. There is a first machine that performs only the first operation of a job and the last machine that performs only the last operation of a job. In a flow shop, the work in a job is broken down into separate tasks called operations and each operation is performed at a different machine. In particular, each operation after the first has exactly one direct predecessor and each operation before the last has exactly one direct successor. A survey of literature has revealed little work reported on the bi-criteria flow shop scheduling problems in fuzzy environment. Prakash [1] developed an algorithm to solve bi-criteria problem more efficiently in reasonable amount of time. Chou and Lee [2] attempt to solve a two machine flow shop bi-criteria scheduling problem with release dates for the jobs to minimize a weighed sum of total flow time and makespan. Gupta et al. [3] proposed the problem of scheduling jobs on parallel identical machines where an optimal schedule is defined as one that gives the smallest maximum tardiness among the set of schedules with optimal total flow time. Sarin and Prakash [4] considered the problem of scheduling the jobs on parallel and identical machines to minimize a primary and the secondary criteria. Peng and Liu [5] developed a methodology for modeling parallel machine scheduling problems with fuzzy processing times and also they presented three novel types of fuzzy scheduling models. Eren and Guner [6] examined the problem of minimizing the total tardiness in a learning effect situation. The concept of learning effects describes the reduction of processing times arising from process repetition. Rahimi-Vahed et al. [7] considered the bi-criteria no-wait flow shop scheduling problem to minimize the weighted mean completion time and weighted mean tardiness simultaneously. Demirel et al. [8] investigated parallel machine scheduling problem in order to minimize total tardiness and developed a genetic algorithm for such problems. Balin [9] proposed a new ‘‘crossover operator’’ and a new ‘‘optimality criterion’’ in order to adapt the GA to non-identical parallel machine scheduling problem. Behnamian et al. [10] presented a min–max multi objective procedure for a dual-objective, namely makespan, sum of the earliness and tardiness of jobs in due window machine scheduling problems simultaneously. Akhshabi et al. [11] provided the bi-criteria to maximize the minimal satisfaction degree with respect to the earliness of jobs and minimal satisfaction degree of tardiness of jobs. Sharma et al. [12] pertained to the bi-criteria scheduling problem on parallel machines to minimizes the weighted flow time. Al-Atroshi et al. [13] offered a hybrid intelligent solution between two approaches: genetic algorithm to arrange the jobs randomly and applied fuzzy logic to build objective function for genetic algorithm. Jadhav and Bajaj [14] developed a method for scheduling the jobs on a single machine to minimize the total penalty cost. Jayanthi and Karthigeyan [15] discussed that scheduling is an allocation of limited resources over a time to perform tasks on machines. Many real-time scheduling problems are both imprecise and uncertain. Fuzzy-set theory has been used to model such systems. Bandyopadhyay and Bhattacharya [16] modified a multi-objective evolutionary algorithm for a parallel machine scheduling problem with multiple objectives. Janaki and Vigithra [17] considered a scheduling involving fuzzy processing time and fuzzy due dates. Nailwal et al. [18] studied a bi-criteria scheduling on parallel machines in fuzzy environment which optimizes the weighted flow time and maximum tardiness. Seema et al. [19] attempt to develop a bi-criteria scheduling on parallel machines in fuzzy environment which optimizes the weighted flow time and total tardiness simultaneously. Recently, Babu et al. [20] adopted methodology to solve n-jobs, m-machines flow shop scheduling problem by using genetic algorithm to get the optimum results of make-span and total tardiness.

In the present work, an attempt is made to optimize a problem of 3 machines for ‘n’ jobs with fuzzy processing time, due time and weightage of jobs by using a genetic algorithm approach. The rest of this paper is organized as: Section 2 provides notations which are used in this paper. In section 3 problems is formulated. Section 4 discusses the genetic operators. Section 5 defines the genetic algorithm proposed to find the optimal ordering for bi-criteria flow shop problem Weighted flow time / Maximum tardiness. In Section 6, computational evaluation is given to support the present genetic algorithm and finally, the conclusion is presented in section 7.

2. NOTATIONS

The following notations are used in this paper:

n : Number of jobs to be scheduled

i : i^{th} Job, $i = 1, 2, 3, \ldots, n$

$\tilde{p}_{i,j}$: Fuzzy processing time of the i^{th} job on j^{th} machine

$p_{i,j}$: Processing time of the i^{th} job on j^{th} machine in crisp one
d_i : Due date of the i^{th} job
c_i : Completion time of i^{th} job
W_i : Weightage of i^{th} job
T_i : Tardiness of the i^{th} job = $\max(0, c_i - d_i)$
$T_{\max}$: Maximum tardiness
k : Machine on which i^{th} job is assigned at the j^{th} position
j : Location of i^{th} job on machine k
WFT : Weighted flow time of jobs
X_{ijk} : 1; if job i is located at position j on k^{th} machine and 0; otherwise
$Ch(i)$: Chromosome number in the population
$F_{Ch(i)}$: Fitness values of each chromosome
$\Pr ob_i$: Probability of selection for i^{th} chromosome.

3. PROBLEM FORMULATION

The following assumptions are made before proceeding with the mathematical formulation in developing the algorithm for bi-criteria flow shop problem on parallel machines:

a) Jobs are available at time zero.
b) Jobs are independent of each other.
c) No pre-emption of jobs is allowed.
d) Machines are identical in all respects.
e) Machines are available all the time.
f) No machine can handle more than one job at a time.

Before formulating the bi-criteria flow shop problem, the mathematical formulation for the single criterion is represented first as given by Prakash [1]. These are as:

3.1 Criterion: Maximum Tardiness ($T_{\max}$)

Tardiness is given by $T_i = \max(0, c_i - d_i)$, where c_i and d_i be the completion and due time of the i^{th} job, respectively. Maximum tardiness is given by $T_{\max} = \max(T_i)$.

The formulation is as follows:

$$Min\, Z = T_{\max}$$

Subject to constraints:

$$\sum_{j=1}^{n}\sum_{k=1}^{n} X_{ijk} = 1 \qquad \forall i \qquad (1)$$

$$\sum_{i=1}^{n} X_{ijk} \le 1 \qquad \forall j, k \qquad (2)$$

$$X_{ijk} \text{ is binary} \qquad \forall i, j, k \qquad (3)$$

$$T_{\max} \ge c_i - d_i \qquad \forall i \qquad (4)$$

along with non-negativity constraints.

3.2 Criterion: Weighted Flow Time (WFT)

The formulation to minimize the weighted flow time is as follows:

$$Min\, Z = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} W_i . X_{ijk}$$

Subject to constraints: set (1), (2) and (3) respectively along with non-negativity constraints.

The formulation of the bi-criteria problem is similar to the single criteria problem but with some additional constraints requiring that the optimal value of the primary objective function is not violated. Here, the present problem is divided into two steps: one is primary in which maximum tardiness of jobs is to be minimized and other is secondary in which weighted flow time of jobs is to be minimized under the objective function value of primary criterion.

3.3 Some Basic Fuzzy Set Theory

Complexity in the real world problems generally arises from uncertainty. From this prospective the concept of fuzzy environment is introduced in the field of scheduling. For example, the processing times of jobs may be uncertain due to incomplete knowledge or uncertain environment which implies that there exist various external sources and types of uncertainty. Fuzzy sets and fuzzy logic can be used to tackle uncertainty inherent in actual scheduling problems. Here, in the present paper, triangular fuzzy membership function is used to represent the uncertainty involved in processing times of jobs.

(a) Triangular Fuzzy Number

A triangular fuzzy number (TFN) is represented with three points as $\tilde{A} = (a, b, c)$, where 'a' and 'c' denote the lower and upper limits of support of a fuzzy set $\tilde{A}$. The membership value of x is denoted by $\mu_{\tilde{A}}(x), x \in R^+$ and defined by:

$$\mu_{\tilde{A}}(x) = \begin{cases} \dfrac{x-a}{b-a}; \ a < x < b \\ \dfrac{c-x}{c-b}; \ b < x < c \\ 0 \quad ; \ otherwise. \end{cases}$$

(b) Defuzzification of Processing Time

In real life situation, the designer would prefer one crisp value for one of the system characteristics rather than fuzzy set. In order to overcome this problem, the fuzzy processing times $\tilde{p}_{i,j}$ are defuzzified by using Robust's ranking method (RRM). In this method, if (a_α^L, a_α^U) is a α- cut for a fuzzy processing time $\tilde{p}_{i,j}$ then its corresponding defuzzified crips value is calculated by the following equation as:

$$p_{i,j} = R(\tilde{p}_{i,j}) = \frac{1}{2}\int_0^1 (a_\alpha^L + a_\alpha^U) d\alpha, where \ a_\alpha^L = a + (b-a)\alpha, \ \alpha_\alpha^U = c - (c-b)\alpha . \qquad (5)$$

4. GENETIC OPERATORS

Genetic algorithm (GA) is one of the common effective meta-heuristic search approaches. Genetic Algorithm is inspired by Darwinian's theory of the survival of the fittest, which means good parents produce better offspring. GA started with a set of solutions called populations. Solutions for one population are taken and used to form a new population. Solutions which are selected to form new population (offspring) are selected according to their fitness. The more suitable they are, the more chances they have to reproduce. This is repeated until some termination conditions for improvement of best solution are satisfied. In this section, we examine various genetic operators for the bi-criteria flow shop scheduling.

A. Initial Population Representation

A population consists of number of individuals being tested and some information about search space. A chromosome in a population is a sequence of gene. Since our bi-criteria flow shop scheduling problem is a sequencing problem of n jobs. Therefore, in order to apply GA to the above described bi-criteria flow shop problem, the structure of a chromosome is expressed as a sequence of the jobs. The length of the chromosomes is equal to the number of jobs. The rules for generating the initial population are described as:

{
Initialize: Number of jobs n, number of machines m
While the number of different chromosomes $Ch(i) \neq n$
{
For $i = 1$ to n
$Ch(i) \leftarrow n - random[0, n-1]$
Select randomly n different chromosomes $Ch(i)$ from the above list in such a way that first gene on each chromosome have different value.
}
}

For example, if bi-criteria flow shop scheduling problem consists of 5 jobs and 3 machines then using the above encoding method, five different chromosomes for the initial population can be:

Population	Ch(1)=	2	1	5	3	4
	Ch(2)=	4	1	3	5	2
	Ch(3)=	3	5	1	2	4
	Ch(4)=	5	4	2	3	1
	Ch(5)=	1	3	4	2	5

B. Fitness Value for each Chromosome

It is well known that choosing a proper fitness function is very important for the effectiveness and efficiency of GA. It provides a means of evaluating the search schedule, and it also controls the selection process. Good fitness functions will help GA to explore the search space more effectively and efficiently. On the other hand, bad fitness functions can easily make GA get trapped in a local optimum solution and lose the discovery power. Each chromosome represents a processing routing of the jobs to the machines in GA. The fitness values of each chromosomes of the population are evaluated by using an objective function. In our case, maximum tardiness and weighted flow time are to be minimized. Fitness value $F_{Ch(i)}$, for chromosome i is computed as:

$$F_{Ch(i)} = \frac{\sum_{i=1}^{n} W_i c_i}{\sum_{i=1}^{n} W_i}, \qquad i = 1, 2, \ldots, n. \tag{6}$$

C. Selection & Reproduction

Selection is the stage of a genetic algorithm in which individual genomes are chosen for the production of the next generation. The selection mechanism determines which individuals are chosen for reproduction and how many offspring each selected individual produces. The main principle of selection strategy is “the better is an individual; the higher is its chance of being parent.” The selection technique for the reproduction used in this paper is based on the probability of the chromosomes. The scheme for reproduction of the children is discussed below:

{
Input: Initial population
Calculate the probability of each chromosome by using the following equation:

$$\Pr ob_i = 1 - \frac{F_{Ch(i)}}{\sum_{i=1}^{n} F_{Ch(i)}}. \quad (7)$$

Delete one chromosome from the population having least probability.
Add one chromosome by copying a chromosome having maximum probability.
Output: New population
}

After applying the selection and reproduction scheme for the above given population, the constructed new generation of the population is:

Before Reproduction	Ch(1)=	2	1	5	3	4
	Ch(2)=	4	1	3	5	2
	Ch(3)=	3	5	1	2	4
	Ch(4)=	5	4	2	3	1
	Ch(5)=	1	3	4	2	5

After Reproduction	Ch(1)=	2	1	5	3	4
	Ch(2)=	4	1	3	5	2
	Ch(3)=	1	3	4	2	5
	Ch(4)=	5	4	2	3	1
	Ch(5)=	1	3	4	2	5

D. Crossover

After the reproduction phase is over, the population is enriched with better individuals. Reproduction makes clones of good strings, but does not create new ones. Crossover operator is applied to the mating pool with a hope that it would create a better string. The aim of the crossover operator is to search the parameter space. In addition, search is to be made in a way that the information stored in the present string is maximally preserved because these parent strings are instances of good strings selected during reproduction. We will not make any change in that chromosome which has been generated by copying a chromosome having maximum probability. The crossover and mutation operations will be applied on remaining (n-1) chromosomes. There are a lot of crossover operators in the literature. But in the present algorithm, *cycle crossover* is used, which is one of the better performers among the others. In the present paper, the crossover probability is assumed to be 1.0. Cycle crossover performs recombination under the constraint that each job comes from one parent or the other. The cycle crossover is illustrated by the following example:

Parent-1	2	1	5	3	4
Parent-2	4	1	3	5	2

Instead of choosing crossing sites, we start from left and choose a job which lies at the first position of the parent-1.

Offspring-1	2	-	-	-	-

Since, we want every job to be taken from one of the two parents, the choice of job 2 from parent-1 means that we must now get job 4 from parent-1 because of the 4 in position of parent-2.

Offspring-1	2	-	-	-	4

This selection in turn requires that we select job 2 from parent-1. This process continues until we are left with the following offspring-1.

Offspring-1	2	-	-	-	4

The selection of 4 means that we should now choose a job 2 from parent-1; however this is not possible; as a 2 having been selected as first job. That we eventually return to the job of origin completes a cycle. This process is giving the operator name. Following the completion of the first cycle, the remaining jobs are filled from the other parent. Completing the example and performing the complementary cross yields the following jobs schedules:

Offspring-1	2	1	3	5	4
Offspring-2	4	1	5	3	2

E. Mutation

After crossover, the strings are subjected to mutation. A simple genetic algorithm treats the mutation only as a secondary operator with the role of restoring lost genetic materials. In the traditional GA, mutation is applied by flipping each elements of the structure from 1 to 0 (or vice versa) with a small probability. In a problem where a chromosome represents a sequence, mutation needs to be defined differently. In the present paper, *interchanging method* for mutation is used. In this two random positions of the string are chosen randomly and their corresponding positions are interchanged. This interchanging mutation method is illustrated by the following example:

Parent	2	**1**	3	**5**	4
		↓		↓	
Offspring	2	**5**	3	**1**	4

5. THE PROPOSED GENETIC ALGORITHM

By combining the above genetic operators, the procedure for finding the optimal ordering of 'n' jobs on 3 machines is developed in the form of algorithm. The outline of the present genetic algorithm can be written as follows:

Step 1 *(Initialization):*

Number of jobs = n, Number of machines = 3.

Step 2 *(Defuzzification):*

Defuzzify the fuzzy processing times $\tilde{p}_{i,j}$ into crisp one $p_{i,j}$ by using Robust's ranking method.

Step 3 *(Arranging the jobs):*

Arrange the jobs in early due dates (EDD) order and find the value of $T_{\max}$.

Step 4 *(Initial population representation):*

Generate an initial population.

Step 5 *(Evaluation of fitness value):*

Evaluate the fitness value for each chromosome.

Step 6 *(Selection and reproduction):*

(a) Make selection of best chromosome according to the selection probabilities

$$\Pr ob_i = 1 - \frac{F_{Ch(i)}}{\sum_{i=1}^{n} F_{Ch(i)}} .$$

(b) Reproduce the population by using present reproduction techniques.

Step 7 *(Crossover):*

Apply the above described cycle crossover operator.

***Step* 8** *(Mutation):*

Apply the above described interchanging mutation operator.

Step 9 *(Termination test):*

If the pre-specified stopping condition is satisfied

Stop the algorithm

Else

Go to ***Step 5***.

Step 10 End

The work flow of the proposed genetic algorithm for the bi-criteria flow shop scheduling problem is shown in the following Figure 1.

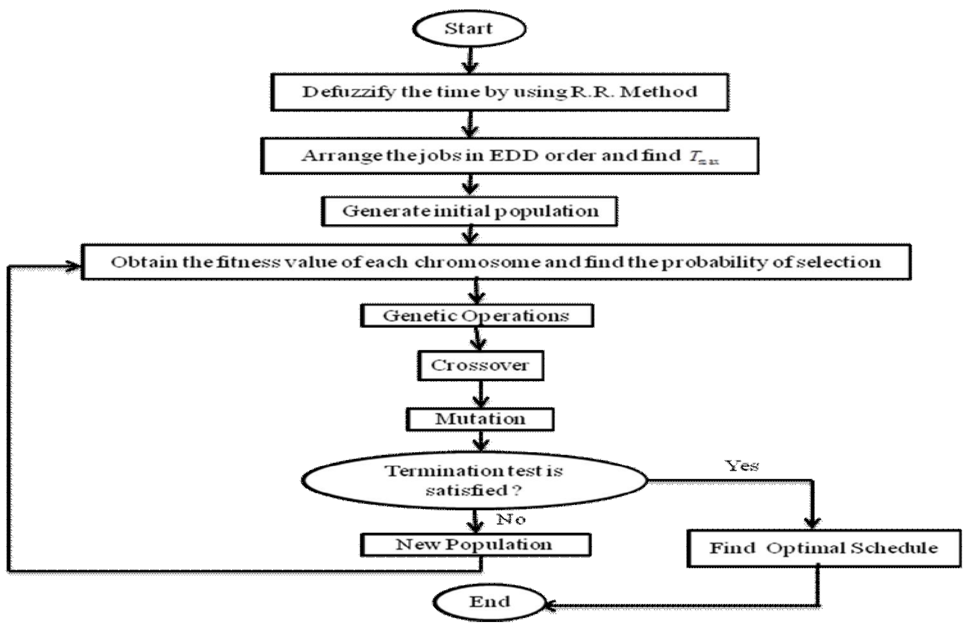

Fig. 1: Work flow of the proposed genetic algorithm

6. COMPUTATIONAL EVALUATION

One example has been illustrated below using the above present genetic algorithm.

Example: Consider an example problem is represented in Table 1. This problem is to execute five jobs on three parallel machines. This table shows original data including jobs with fuzzy processing times, due times and jobs weightage on three parallel machines.

Table 1: Jobs with fuzzy processing time

Jobs	1	2	3	4	5
Processing Time ($\tilde{p}_{i,j}$)	(8, 9, 10)	(15, 16, 17)	(8, 9, 10)	(5, 6, 7)	(10, 11, 12)
Due Time (d_i)	12	18	17	10	14
Weightage (W_i)	3	4	2	5	1

The RRM of processing time of given jobs as per the *Step 2* using Equation 5 is as follows:

Table 2: Jobs with crisp processing time

Jobs	1	2	3	4	5
Crisp Processing Time $p_{i,j}$	9	16	9	6	11
Due Time (d_i **)**	12	18	17	10	14
Weightage (W_i **)**	3	4	2	5	1

On arranging the jobs as per *Step 3*, the order of jobs becomes $\{4,1,5,3,2\}$. The in-out, i.e. the flow of jobs on machines for EDD order is as shown in Table 3.

Table 3: Jobs scheduling by EDD order

Jobs	4	1	5	3	2
M_1	0-6			6-15	
M_2		0-9			9-25
M_3			0-11		
Weightage (W_i **)**	5	3	1	2	4
Due Time (d_i **)**	10	12	14	17	18
Tardiness (T_i **)**	–	–	–	–	7

After applying *Steps 4 to 10*, we get the optimal ordering of the jobs as $\{4,2,1,3,5\}$. The flow table for the sequence $\{4,2,1,3,5\}$ of the jobs is represented in Table 4.

Table 4: Scheduling of jobs by present genetic algorithm

Jobs	4	2	1	3	5
M_1	0-6			6-15	
M_2		0-16			
M_3			0-9		9-20
Weightage (W_i **)**	5	4	3	2	1
Due Time (d_i **)**	10	18	12	17	14
Tardiness (T_i **)**	–	–	–	–	6

Here, $T_i = 6 < T_{\max}$, i.e. there is no late job that violates the primary criterion of $T_{\max}$. Hence the solution optimizing the bi-criteria minimum $WFT / T_{\max}$ is obtained.

The minimum weighted flow time is $\dfrac{6\times5+16\times4+9\times3+15\times2+20\times1}{5+4+3+2+1} = \dfrac{171}{15} = 11.4$ units

and maximum tardiness $T_{\max}$ = 7 units.

Thus, the optimal sequence of jobs on machines is $4\to2\to1\to3\to5$ with minimum WFT as 11.4 units and minimum $T_{\max}$ as 7 units.

We conducted an experiment by solving randomly generated flow shop problems of different sizes with the following specifications:

Table 5: Specification of the Problem

Parameter	Values
Number of Jobs	5-15
Number of Machines	2-3
Processing Time of the Jobs on a Machine	Fuzzy Numbers
Due Time of a Job	10-20
Weightage of Jobs	1-8
Crossover probability	1.0
Crossover Operator	Cycle Crossover
Mutation Probability	0.05
Mutation Operator	Interchanging Method
Number of iterations	40

More than 50 independent problems for different combinations of jobs and machines are solved by the present proposed algorithm. The present algorithm has been coded in C++. To check the effectiveness of the proposed genetic algorithm, we have compared it with four well known algorithms in the literature. Four problems were taken from the literature to demonstrate the present genetic approach. A comparison of the results is given in Table 6.

Table 6: Comparison of the results

Problem	Reference	Objective function	Result	Proposed genetic algorithm
1.	**Sharma et al. [12]**	Minimum weighted flow time	12.2	11.4
		Maximum tardiness	8.3	7
2.	**Nailwal et al. [18]**	Weighted flow time	201	189
		Maximum tardiness	8.3	8
3.	**Seema et al. [19]**	Minimum weighted flow time	11.80	10.86
		Total tardiness	25	21
4.	**Babu et al. [20]**	Make-span	695	695
		Total tardiness	4	4

The results of the Table 6 show that the present genetic algorithm is quiet better in comparison to the other algorithms for finding the optimal scheduling of the jobs to the machines.

7. CONCLUSIONS

In the present paper, an attempt is made to find an optimal ordering of 'n' jobs on 3 machines involving processing times, due time and weightage of jobs by using genetic algorithm approach. We have developed an efficient GA based algorithm for solving the bi-criteria flow shop scheduling on parallel machines which is easily implementable and performs quiet effectively. Present algorithm protects the best schedule of the jobs which has the minimum weighted flow time at each iteration. To test the efficiency of the proposed algorithm, a large number of problems are solved and it is found that the algorithm is workable in all the cases. The present proposed algorithm is also compared with four well known algorithms in the literature. The solutions produced by the proposed algorithm are statistically significantly better than those produced by these four algorithms.

REFERENCES

1. D. Prakash (1997), "Bi-criteria Scheduling Problems on Parallel Machines", Ph.D. Thesis, University of Blacksburg, Virginia.

2. F. D. Chou and C. E. Lee (1999), “Two-Machine Flow shop Scheduling with Bi-criteria Problem”, Computers & Industrial Engineering, Vol. 36, pp. 549-564.
3. J. N. D. Gupta, A. J. Ruiz-Torres and S. Webster (2003), “Minimizing Maximum Tardiness and Number of Tardy Jobs on Parallel Machines Subject to Minimum Flow-Time”, Journal of the Operational Research Society, Vol. 54, No. 12, pp.1263-1274.
4. S. C. Sarin and D. Prakash (2004), “Equal Processing Time Bi-criteria Scheduling on Parallel Machines”, Journal of Combinatorial Optimization, Vol. 8, pp. 227-240.
5. J. Peng and B. Liu (2004), “Parallel Machine Scheduling Models with Fuzzy Processing Times”, Information Sciences, Elsevier, Vol. 166, pp. 49-66.
6. T. Eren and E. Guner (2007), “Minimizing Total Tardiness in a Scheduling Problem with a Learning Effect”, Applied Mathematical Modelling, Elsevier, Vol. 31, pp. 1351-1361.
7. A.R. Rahimi-Vahed, B. Javadi, M. Rabbani and R. Tavakkoli-Moghaddam (2008), “A Multi-Objective Scatter Search for a Bi-criteria No-wait Flow Shop Scheduling Problem”, Engineering Optimization, Taylor & Francis, Vol. 40, No. 4, pp. 331-346.
8. T. Demirel, V. Ozkir, N. C. Demirel, and B. Tasdelen (2011), “A Genetic Algorithm Approach For Minimizing Total Tardiness in Parallel Machine Scheduling Problems”, Proceedings of the World Congress on Engineering, July 6-8, London, U.K., Vol. 2.
9. S. Balin (2011), “Non-identical Parallel Machine Scheduling using Genetic Algorithm”, Expert Systems with Applications, Vol. 38, pp. 6814-6821.
10. J. Behnamian, M. Zandieh and S. M. T. Fatemi Ghomi (2011), “Bi-objective Parallel Machines Scheduling with Sequence-dependent Setup Times using Hybrid Metaheuristics and Weighted Min–Max Technique”, Soft Computing, Springer, Vol. 15, pp. 1313-1331.
11. M. Akhshabi, Mostafa Akhshabi and J. Khalatbari (2012), “Bi-criteria Flow Shop Scheduling With Fuzzy Simulated Annealing Algorithm”, African Journal of Business Management, Vol. 6, No. 25, pp. 7478-7488.
12. S. Sharma, D. Gupta and Seema (2013), “Bi-criteria scheduling on Parallel Machines: Total Tardiness and Weighted Flow Time in Fuzzy Environment”, International Journal of Mathematics in Operational Research, Vol. 5, No. 4, pp. 492-507.
13. M. Al-Atroshi, S. T. Azez and B. S. Bhnam (2013), “An Effective Genetic Algorithm for Job Shop Scheduling with Fuzzy Degree of Satisfaction”, International Journal of Computer Science Issues, Vol. 10, No. 1, pp. 180-185.
14. V. S. Jadhav and V. H. Bajaj (2013), “Fuzzy Average High Ranking Method for Solving Single Machine Scheduling Problem”, International Journal of Statistika and Mathematika, Vol. 6, No. 2, pp. 56-59.
15. S. E. Jayanthi and S. Karthigeyan (2013), “One-Operation Jobs Scheduling under Fuzziness – A Review”, International Journal of Fuzzy Mathematics and Systems, Vol. 3, No. 3, pp. 251-257.
16. S. Bandyopadhyay and R. Bhattacharya (2013), “Solving Multi-Objective Parallel Machine Scheduling Problem by a Modified NSGA-II”, Applied Mathematical Modelling, Vol. 37, pp. 6718- 6729.
17. E. Janaki and. R.Vigithra (2015), “One - Machine Scheduling Problem with No Common Due Dates under Fuzzy Environment”, IOSR Journal of Mathematics, Vol. 11, No. 4, pp. 61-64.
18. K.K. Nailwal, D. Gupta and S. Sharma (2015), “Fuzzy Bi-criteria Scheduling on Parallel Machines involving Weighted Flow Time and Maximum Tardiness”, Cogent Mathematics, Vol. 2, pp. 1-10.
19. Seema, S. Sharma and G. Khanna (2015), “Bi-criteria Scheduling on Parallel Machines under Fuzzy Processing Time”, 21st International Congress on Modelling and Simulation, 29 Nov to 4 Dec, Gold Coast, Australia.
20. P. M. Babu, B. V. H. Sai and A. S. Ready (2015), “Optimization of Make-span and Total Tardiness for Flow Shop Scheduling using Genetic Algorithm”, International Journal of Engineering Research and General Science, Vol. 3, No. 3, pp. 195-199.