I.J. Intelligent Systems and Applications, **2015, 04, 54-61**
Published Online March 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2015.04.08

Hybrid Multi-Objective Particle Swarm Optimization for Flexible Job Shop Scheduling Problem

S. V. Kamble
PG Student, Department of Computer Science & Engineering, Rajarambapu Institute of Technology Sakharale, MS, India
E-mail: sach06jan@gmail.com

S. U. Mane
Department of Computer Science & Engineering, Rajarambapu Institute of Technology Sakharale, MS, India
E-mail: sandip.mane@ritindia.edu

A. J. Umbarkar
Department of Information Technology, Walchand College of Engineering Sangli, MS, India
E-mail: anantumbarkar@rediffmail.com

Abstract— Hybrid algorithm based on Particle Swarm Optimization (PSO) and Simulated annealing (SA) is proposed, to solve Flexible Job Shop Scheduling with five objectives to be minimized simultaneously: makespan, maximal machine workload, total workload, machine idle time & total tardiness. Rescheduling strategy used to shuffle workload once the machine breakdown takes place in proposed algorithm. The hybrid algorithm combines the high global search efficiency of PSO with the powerful ability to avoid being trapped in local minimum of SA. A hybrid multi-objective PSO (MPSO) and SA algorithm is proposed to identify an approximation of the pareto front for Flexible job shop scheduling (FJSSP). Pareto front and crowding distance is used for identify the fitness of particle. MPSO is significant to global search and SA used to local search. The proposed MPSO algorithm is experimentally applied on two benchmark data set. The result shows that the proposed algorithm is better in term quality of non-dominated solution compared to the other algorithms in the literature.

Index Terms— Particle Swarm Optimization; Simulated Annealing; Multi Objective Optimization, Job Shop Scheduling, Metaheurestic

I. Introduction

The job shop scheduling problem (JSSP) is considered the most challenging one which has been the subject of many research studies for consumer demand and development in the production system during the recent decades. The problem is described simply as follows: given n jobs to be processed on m machines. Each job consists of a predetermined sequence of task operations, each of which requires processing without interruption for a given period of time on a given machine. The Flexible Job-shop Scheduling Problem (FJSSP) is an extension of the classical JSSP, which allows an operation to be processed by any machine from a given set or at least one operation may not process on all machines. So, FJSSP is more complex than JSSP because of additional need to determine the assignment of operations to machines [3]. FJSSP is closer to the real production situations, for example, it is used in flexible manufacturing systems. A flexible manufacturing system consists of several CNC machines. A CNC machine is a multi-tasking machine. So flexible job shop scheduling is applicable for flexible manufacturing systems [5].

Job Shop Scheduling (JSSP) is one of the most complex scheduling problems related to manufacturing industries. However, FJSSP is also an NP-hard problem as the number of jobs increases; it becomes more difficult to obtain the optimal schedule in short time. The problem of scheduling jobs in FJSSP could be decomposed into two sub-problems: Assignment Problem is to select a machine from several available machines for each operation. The second sub problem, sequence problem, is to identify a sequence of all operations on each machine. There are two different methods are used to solve this problem. The first approach, Hierarchical approach used to solve two sub problems separately. The Second approach, integrated approach considers two sub problems simultaneously [18].

However, most of the research used one objective or three objectives to be optimized with hierarchical approach. In this paper, we propose an integrated approach based on hybridization of particle swarm optimization and simulated annealing algorithm for solving the flexible job-shop scheduling problem with five objectives optimized simultaneously. Also rescheduling strategy is implemented. PSO allows an extensive search of solution space for global search while the simulated annealing is used for local search a hybrid multi-objective particle swarm optimization (MPSO) and simulated annealing (SA) algorithm is used to identify an approximation of the Pareto front for FJSSP. Hybrid MPSO is significant for global search and SA is used for local search.

The rest of the paper is organized as follows: Literature review is presented in section II, Problem description and formulation is presented in section III, section IV describes the Hybrid MPSO proposed for the problem under discussion with numerical illustration, section V presents the performance comparison and results of the proposed Hybrid MPSO and section VI concludes giving scope for future research.

II. Literature Review

Many researches have been done on FJSSP scheduling and several methods that include integrated approach and hierarchical, have been developed to solve FJSSP. Hierarchical approaches consist of decomposing the original problem in order to reduce its complexity. This type of approach is natural for FJSSP since the routing and the scheduling sub-problem can be separated.

Brandimarte [1] was the first to apply the decomposition approach into the FJSSP. He solved the routing sub-problem by using some existing dispatching rules and then focused on the scheduling sub-problem; it was solved by using a tabu search heuristic. Paulli [2] applied hierarchical approach.

Hurink [3] represented the FJSSP as a disjunctive graph model and proposed a hierarchical approach based on TS to solve the problem for minimum makespan time criterion.

Dauzere-Peres and Paulli [4] presented a new disjunctive graph model to represent the FJSSP problem and proposed an integrated approach based on TS to solve FJSSP.

Chen [5] also applied genetic algorithm to solve FJSSP and introduced a new coding for each solution and different crossover and mutation operators to minimize the makespan for all jobs.

Mastrolilli and Gamberdella [6] improved the TS approach proposed by Dauzere-Peres and Paulli and presented two new neighborhood functions to solve FJSP instances.

Most of the above-mentioned research is considered hierarchical or integrated approach with a single objective optimization in FJSSP. However, in modern production systems of FJSSP, more than one objective is required to be optimized simultaneously. Therefore, some researchers have considered the optimization of more than one objective in FJSSP.

Sha, Lin [7] proposed evolutionary PSO technique to solve the JSSP with multiple objectives. They used a diversification strategy with modified the particle position representation, particle movement, and particle velocity.

Xia and Wu [8] developed an easily implemented approach for the multi-objective flexible JSSP based on the combination of PSO and SA. A particle swarm optimization (PSO) is to use assign each operation to appropriate machine and simulated annealing (SA) to sequence of operation on the machine. PSO provides an initial solution for SA during the hybrid search process.

Kacem [9] he has implemented a technique that solves the assignment and job-shop scheduling problems with total or partial flexibility. He has used two approaches, localization (AL) and evolutionary. Localization used to solve the problem of resource allocation and build an ideal assignment model .Evolutionary approach controlled by the assignment model.

Gao [10] developed a hybrid genetic algorithm (GA) for the FJSSP. The GA uses two vectors to represent solutions. Advanced crossover and mutation operators are used to adapt to the special chromosome structure and the characteristics of the problem. Individuals of GA are first improved by a variable neighborhood descent (VND) which involves two local search procedures: local search of moving one operation and local search of moving two operations.

Li [11] proposed hybrid algorithm combining PSO and tabu search (TS) to solve (JSSP) with fuzzy processing time. PSO used for the global search, i.e. the exploration phase, while TS used for the local search i.e. the exploitation process. The global best particle is used to direct other particles to optimal search space.

Tavakkoli-Moghaddamn [12] proposed new multi objective Pareto archive particle swarm optimization (PSO) algorithm combined with genetic operators as variable neighborhood search (VNS). Character of scatter search (SS) to select new swarm in every iteration order to find Pareto optimal solutions for the given problem. Pareto archive PSO has been combined with genetic operators to update and VNS to improve particles.

Zhang, Shao[13] developed a particle swarm optimization (PSO) algorithm and a tabu search (TS) algorithm are combined to solve the multi-objective FJSSP with several conflicting and incommensurable objectives.

Saidi-Mehrabad, Fattahi [14] proposed TS algorithm for FJSSP, in this technique the initial feasible solution of jobs and operations sequences is first generated and then the algorithm searches for the best choice of the machine's alternative for this job and operation sequence.

Carlo, Raquel, Prospero, Naval [15] proposed an approach that extends the Particle Swarm Optimization (PSO) algorithm to handle Multi-objective optimization problems by incorporating the mechanism of crowding distance computation into the algorithm of PSO.

Lie [16] presented a PSO for the multi-objective JSSP with objective minimize makespan and total job tardiness simultaneously. Job-shop scheduling problem can be converted into a continuous optimization problem by constructing the corresponding relationship between a real vector and a chromosome obtained using the priority rule-based representation method. The global best position selection is combined with crowding-measure-based archive maintenance to design a Pareto archive PSO. That algorithm is capable of producing a number of high-quality Pareto optimal scheduling plans.

Li, Pan, Xie, Wang [17] proposed a hybrid Pareto-based artificial bee colony (HABC) algorithm for solving the multi-objective flexible job shop scheduling problem.

Sahana, Jain [28] they are analyzed Indian train scheduling from source to destination with the iteration wise conversion process with Ant colony optimization.

Guezouri, Houacine[29] they proposed an algorithm based on the principle of clonal selection and affinity maturation mechanism in an immune response used to solve the Hybrid Flow Shop (FSH) scheduling problem.

Recently, Shao [18] developed a novel hybrid DPSO and SA for FJSSP. The proposed algorithm is based on the integrated approach of multi-objective optimization developed by Shao [18] adding with more objective optimization with rescheduling strategy.

III. FJSSP Problem Formulateion

Problem Description:

In flexible job-shop problem, a set of *n* jobs is considered to process on a set of *m* machines. Each job, denoted by J_i $(1 \leq I \leq n)$ has a predefine sequence n_j of operations. Each operation O_{ij} is processed by any one machine from a set M_{ij}, which is the subset of machines that can perform O_{ij}. The processing time of O_{ij} on machine k $(1 \leq k \leq m)$ is fixed and is denoted by P_{ijk}. The FJSSP is designed to assign all the operations of the jobs to available machines to identify their starting and completion time and is aimed to obtain an optimal schedule with some objectives.

Assumptions:

- All job released at time 0.
- All machines available at time 0.
- Only one job can be carried on one machine at one time.
- Once an operation starts, it cannot be terminated before it finishes.
- Order constrains only exist in operation on the same job.

Objective

The objectives are to find an assignment and a schedule to minimize following objective simultaneously.

1) Ms: Makespan, i.e., the maximal completion time of machines or jobs.
2) Mw: Machine workload, i.e., the maximum processing time spent on any machine; this objective is used to keep the balance of workload among different machines and may help to avoid too much work scheduling on a certain machine.
3) Tw: The total workload of machines, which is defined as total processing time over all machines.
4) Ti: Total idle time, which is defined idle time of machines.
5) Tr : Total Tardiness, which is defined as lateness of jobs.

IV. Hybrid PSO AND SA FOR FJSSP

A. PSO algorithm

PSO is an evolutionary computation technique proposed by Kennedy and Eberhart [20]. A PSO algorithm consists of behavior of flying birds and it means that they exchange of information to solve optimization problems. PSO has been introduced as an optimization technique in real-number spaces. Typical examples include problems that require ordering and route planning in scheduling, to solve FJSSP for multi-objective converting the continuous domain to the discrete domain for PSO. In this strategy the velocity and displacement of particles need to be redefined for FJSSP.

Standard PSO

PSO is similar to the evolutionary algorithm in that the system is initialized with a population (“Swarm”) of random solutions. Each individual or potential solution, called a particle, flies in the D-dimensional problem space with a velocity that is dynamically adjusted according to the flying experience of the individual and its colleagues. Each particle remembers the best position that it has found so far during the search process personal best (*pbest*), and knows the best position of the swarm global best (*gbest*). Therefore, each particle interacts with other and every particle in the swarm tries to gradually move toward the promising areas of the search space and in this way an optimum solution is found.

The global model of equations is given below:

$$M_{id} = W \bullet M_{id} + C1 \bullet Rand\,() \bullet (P_{id} - N_{id}) + C2 \bullet Rand\,() \bullet (P_{gd} - N_{id}) \quad (1)$$

$$N_{id} = N_{id} + M_{id} \quad (2)$$

Where, M_id the velocity for particle *i* represent the distance to be traveled by particle *I* from its current position. N_{id} represents the particle position; P_{id} represents “*pbest*” the local best solution of i_{th} particle’s best previous position. P_{gd} is gbest the global best solution represents the best position among all particles in the swarm.

W is the interia weight it regulates the trade of between the global exploration and local exploitation abilities of the swarm. The acceleration constants *C1* and *C2* represent the weight of the stochastic acceleration terms that pull each particle toward “*pbest*” and “*gbest*” positions. *Rand* ( ) are two random functions with range (0, 1).

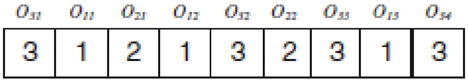
The first part Eq.1 represents the inertia of previous velocity; the second part is the “cognition” part, which represents individuals thinking independently; and the third part is the “social” part, represents cooperation among the particles.

B. Particle representation:

In the FJSSP consist of two sub problem as assignment and sequence problem. So that particle represents in two vector one vector represents the operation permutation solution and second vector represents machine allocation Solution. Both vectors are *I*-dimensional vector, where *I* is the total number of all operations in FJSSP.

Permutation vector for Operations:

Operation permutation specifies the sequence of operation all operations from the same job occupy the same index in present vector.



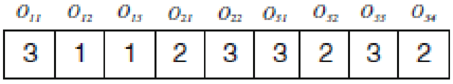
O_{31}	O_{11}	O_{21}	O_{12}	O_{32}	O_{22}	O_{33}	O_{13}	O_{34}
3	1	2	1	3	2	3	1	3

Fig. 1. Operation Allocation Vector

In the above vector first element 3 specifies operation 1 for the job 3 that is $O_{31.}$ Similarly 1 specifies operation 1 of the job 1 that is $O_{11.}$

Allocation Vector for Machines:

Machine allocation vector has same length as operation vector it shows the allocation of specific job on a particular machine as per shortest processing time rule (SPT). In machine allocation vector O_{11} indicates operation 1 of job 1 processed on machine 3 as per SPT.



O_{11}	O_{12}	O_{13}	O_{21}	O_{22}	O_{31}	O_{32}	O_{33}	O_{34}
3	1	1	2	3	3	2	3	2

Fig. 2. Machine Allocation Vector

C. Particle Decoding for FJSSP

For FJSSP particle is decoded using active schedule follows. Staring time S_{ij} competition time C_{ij} and processing time p_{ij}, M_k denotes completion time of last operation in the schedule.

1. If the operation is the first operation for a job *I* then $S_{ij}=C_{ij}=0$

2. Processing time of the next operation consists of $C_{ij}=S_{ij}+$ pig

3. If number of operations assigned to the machine, find the idle interval on machine between two successive operations.

D. Update of pbest()

In the proposed algorithm, we have taken consideration of *Pbest* of particle as non-dominated solution which represents a personal best of a particle with no other objective are dominating. *Pbest* value stored naturally in them without the need of an external archive. The maximal archive size of the proposed algorithm is set equal to the number of all particles in the swarm. For every particle in the swarm, a displacement of a particle generates a new solution Hn+1, and the relationship between Hn+1 and Hpbest determines the update of Hpbest. With the role of domination, there are two cases between Hn+1 and Hpbest after a displacement for each particle has occurred and these different cases are as follows.

1) If Hn+1 dominates Hpbest, Hpbest is replaced by the new solution Hn+1.

2) If Hpbest dominates Hn+1, Hpbest is unchanged

E. Selection of gbest()

In single objective optimization there is only one optimal solution where as in multi-objective optimization aim to find the Pareto set instead of a single solution. In the proposed algorithm, gbest is considered as a leader for each particle which is randomly selected from pbest set with the selection probability proportional to the fitness value.

F. Fitness Function

Fitness is used as the performance evaluation of particles in the swarm. Fitness is usually represented with a function f: S→R (where S is the set of candidate schedules, and R is the set of positive real values). In the current algorithm, the current position Hn and the best position founded so far Hpbest are considered as two attributes of the particle. Each particle in the proposed algorithm is evaluated using Hpbest instead of Hn and the fitness of each particle is computed through the Pareto ranking and crowding distance method.

Crowding distance

The crowding distance is the distance between two particles and it is used to compute the degree of crowd of particles in an objective space.

$$Rij=|Ms(i)-Ms(j)|+|Mw(i)+Mw(j)|+ ||Tw(i)-w(j)||+||Ti(i)-T(j)||+||Tr(i)-Tr(j)| \tag{3}$$

$$Ed(i)=Min\{Ri1,Ri2,....Rik,...Rin\} \tag{4}$$

Pareto Ranking

Ranking of particle consist with non-dominated solution in the objective space.

$$Fit=exp\ (-m + n)\ (rank-1+exp(-m+n)Ed))/I \tag{5}$$

Where, *I* is the total number of operations, *n* is the number of jobs, *m* is the number of machine and *Ed* is crowding distance of the particle.

Diversification strategy:

All the particles have the same non-dominated solutions; they will be trapped in local optima, to prevent this displacement of particle to keep the non-dominated solutions differently. Once any new solution is generated by particles, the non-dominating solution set will be updated in these three situations:

1) The solution of the particle dominates the gbest solution, assign the particle solution to the gbest.
2) The solution of the particle equals to any solution in the non-dominated solution set, replace the non-dominated solution with the particle solution.
3) The solution of the particle is dominated by the worst non-dominated solution and not equal to any non-dominated solution, set the worst non-dominated solution equal to the particle solution.

Displacement of particle:

The displacement of each particle depends on its *pbest* and *gbest,* in which the velocity of particle represents the probability of a particle to change towards *pbest* and *gbest*. The displacement of each particle contains two parts, i.e., operation permutation and machine allocation.

```
Procedure MPSO
Begin
Step I Initialization
1. PSO
a. Initialize swarm, including swarm size, particle position and velocity.
b. calculate the three objects (Ms, Mw, Tw, Ti, Tr), Ranking all particles with Pareto Set concept
2. SA
c. Determine the initial temperature level. T, Tend, B
Step II Computation
PSO
While (T> Tend) for each particle do
a. Particle displacement
b. Calculate the three objects (Ms, Mw, Tw, Ti, Tr), Ranking all particles with Pareto Set concept. Calculate the Crowding distance Cd according to Eq. (3) on each Pareto Rank. If necessary, update gbest.
c. evaluate each particle according Eq. (4).
    2. SA begin
Set k=1;
While ( k<=Km ) for each particle do
a. Randomly choose a neighbor
b. Accepting neighbor if find an optimal solution with respect to the current one, update it
k=k+1;
End while
Update T=T Kb;
SA end
Step III:
Output optimization result
 End While
```

Fig. 3. Framework of Hybrid MPSO

G. SA Algorithm

Starting from an initial solution, SA generates a new solution S’ in the neighborhood of the original solution S. Then, the change of objective function value, $\Delta = f(S') - f(S)$, is calculated. For a minimization problem, if $\Delta < 0$, the transition to the new solution is accepted. If $\Delta \geq 0$, then the transition to the new solution is accepted with probability, usually denoted by the function $\exp(-\Delta/T)$, where T is a control parameter called the temperature. The SA algorithm generally starts from a high temperature, and then the temperature is gradually lowered. At each temperature, a search is carried out for a certain number of iterations, called the epoch length. When the termination condition is satisfied, the algorithm will stop [21].SA acts as a local search method under the framework of the proposed hybrid MPSO.

In the proposed algorithm, the number of iterations for PSO is defined by the initial temperature level T0, constant Kb and the final temperature (Tend). The framework of the proposed hybrid MPSO algorithm is illustrated in Fig. 3. The temperature decreases with ratio kB in an iteration of the algorithm and the algorithm terminates once the temperature is less than the final temperature (Tend).

H. Rescheduling of FJSSP

In the proposed algorithm rescheduling is done. In this method, it finds machine breakdown randomly and shuffle the Workload among available machine. The workload of breakdown machine is transferred to other machines to keep objectives of FJSSP will minimize simultaneously [23].

Table 1. Comparision of hybrid MPSO to other algorithm

Size	Objective	AL+CGA[9]	PSO+SA[8]	SM[21]	HGA[10]	PSO+TS[19]	AIA[17]	HDPSO[18]	**Proposed HMPSO**	**Proposed Rescheduling**
8 x 8	Ms	15,16	15,16	16	14	15	14	14	14	17
	Mw	-	12,13	13	12	12	12	12	12	16
	Tw	79,75	75,73	73	77	75	77	77	77	83
	Ti	-	-	-	-	-	-	-	21	22
	Tr	-	-	-	-	-	-	-	19	37
10 x 10	Ms	8,7	7	8	7	7	7	7	7	8
	Mw	7,5	6	7	5	6	6	5	6	8
	Tw	41,45	44	41	43	43	43	43	43	50
	Ti	-	-	-	-	-	-	-	11	10
	Tr	-	-	-	-	-	-	-	20	30
15 X 15	Ms	23.24	12	11	12	11	12	11	11	12
	Mw	11.11	11	11	11	11	11	11	11	13
	Tw	95.91	91	91	91	91	91	91	91	97
	Ti	-	-	-	-	-	-	-	20	27
	Tr	-	-	-	-	-	-	-	38	40

V. Performance Comparison And Results

Effectiveness and performance of the proposed algorithm are checked on the two benchmark data set of job-shop instances with different sizes. The proposed algorithm is coded in Java and implemented on a personal computer with 3.2 GHz and 4 GB RAM. The final temperature (*Tend*) is set to 0.01 and *Kb* is set to0.98.

Population size is set to 100 for 4 ×5,150 for 7×10,200 for 8x8, 300 for10x10 and 400 for 15x10 respectively. The non-deterministic nature of our algorithm makes it necessary to carry out multiple runs on the same problem instance in order to obtain meaningful results. Two benchmark sets are considered.

Firstly, the proposed algorithm is compared with some well-known algorithms in literature on Kacem data. An Instance of P-FJSP that consists of 27 operations. Problem 10×10 and Problem 15×10 belong to T-FJSP it contains 30 and 56 operations respectively. The proposed algorithm is compared with other famous algorithms in literature. These algorithms include “AL+CGA” of Kacem et al. [9], “PSO+SA” of Xia and Wu [8], “hGA” of Gao et al. [10], “SM” of Xing et al. [21], “hybrid PSO-TS” of Zhang et al. [19] , “AIA” of Bagheri et al. [17] and “HDPSO” of Xinku Shao et.al [18].

Brandimarte data set: BR data composed of 10 problems from Brandimarte [1]. In these problems, the number of jobs ranges from 10 to 20, the number of machines ranges from 4 to 20, and the average of available machines for an operation (flex) ranges from 1.43 to 4.10.

Table 2. Performance comparison between proposed hybrid MPSO and MOPSO+LS ,Fl+EA on kacem data.

Size	Objective	FL+EA[9]	MOPSO+Ls[9]	**Proposed HMPSO**	**Proposed Rescheduling**
4 x 5	Ms	18,18	16,16	11	14
	Mw	8,7	8,7	9	12
	Tw	32,33	32,33	32	38
	Ti	-	-	10	11
	Tr	-	-	6	16
10 x 7	Ms	16,15	16,15	12	14
	Mw	12,11	12,11	12	14
	Tw	60,61	61,60	63	72
	Ti	-	-	7	7
	Tr	-	-	42	54

Table 3. Performance comparison between proposed hybrid MPSO and (AIA, SM) and hMPSO

	Size	Objective	AIA[17]	SM[27]	HDPSO[18]	**Proposed HMPSO**	**Proposed Rescheduling**
MK01	10x6	Ms	40	42	40	40	42
		Mw	36	42	36	36	37
		Tw	171	162	167	163	171
		Ti	-	-	-	33	35
		Tr	-	-	-	72	81
MK01	10x6	Ms	26	28	27	27	29
		Mw	26	28	27	26	32
		Tw	154	155	145	144	149
		Ti	-	-	-	35	46
		Tr	-	-	-	85	91
MK03	15x8	Ms	204	204	210	204	220
		Mw	204	204	210	210	254
		Tw	1207	854	848	852	961
		Ti	-	-	-	90	95
		Tr	-	-	-	145	181
MK04	15x8	Ms	60	68	61	61	65
		Mw	60	67	60	60	74
		Tw	403	372	366	365	401
		Ti	-	-	-	85	89
		Tr	-	-	-	121	135

Table 4. Performance comparison between proposed hybrid MPSO and (AIA, SM) and hMPSO Cont.

MK05	15x4	Ms	173	177	173	175	185
		Mw	173	177	173	173	191
		Tw	686	702	683	682	700
		Ti	-	-	-	80	85
		Tr	-	-	-	111	121
MK06	10x15	Ms	63	75	62	62	68
		Mw	56	67	58	58	64
		Tw	470	431	412	412	451
		Ti	-	-	-	75	78
		Tr	-	-	-	94	101
MK07	20x5	Ms	140	150	141	140	155
		Mw	140	150	141	141	161
		Tw	695	717	692	693	703
		Ti	-	-	-	45	52
		Tr	-	-	-	52	74
MK08	20x10	Ms	214	227	211	211	241
		Mw	203	221	207	207	250
		Tw	2121	1989	1998	1995	2014
		Ti	-	-	-	98	142
		Tr	-	-	-	143	175

VI. CONCLUSION

There is a large amount of research into the FJSSP, most of this has focused on minimizing the maximum completion time (i.e. makespan). There exist other objectives in the real world, such as the minimization of machine idle time that might help improve efficiency and reduce production costs, Total Tardiness specify the lateness of the job.

The hybrid MPSO method is used to solve the FJSSP with multiple objectives and rescheduling strategy, PSO algorithm is significant for continuous problems and FJSSP is by discrete nature. Therefore, the displacement of a particle is redefined. In the proposed algorithm, Non-dominated solutions are stored naturally by using *pbest.* Simulated Annealing is used for local search in the algorithm, *pbest* of particles is used to store the best solutions found from SA. A new selection mechanism is developed based on the Pareto ranking scheme.

Experimental Results show that the proposed algorithm gives better results as compared to the algorithms given in literature. On future scope, researchers can focus on optimizing some other objectives of FJSSP. Furthermore, some other meta- heuristic algorithms could be experimented to solve the FJSSP.

ACKNOWLEDGMENT

The authors are thankful to the anonymous reviewers as well as the editors for their valuable comments and suggestions which have led to improve the quality of the paper. Our special thanks to S. Xinyu, W. Liu, L. Qiong and Z. Chaoyong for their work on Job shop scheduling using particle swarm optimization Algorithm published in various journals and conferences.

REFERENCES

[1] P. Brandimarte, “Routing and scheduling in a flexible job shop by tabu Search,” Annals of Oper. Res., 1993, Vol. 41, pp. 157-183.

[2] J Paulli, “A hierarchical approach for the FMS scheduling problem,” Eur J. Oper. Res., 1995, Vol. 86(1), pp. 32–42.

[3] J. Hurink , B. Jurisch and M. Thole, “Tabu search for the job-shop scheduling with multi-purpose machines,” OR Spektrum, 1994, Vol. 15, pp.205-215.

[4] S. Dauzere-Peres, J. Paulli, “An integrated approach for modeling and solving the general multiprocessor job-shop scheduling problem with tabu search,” Annals of Oper. Res., 1997, Vol. 70, pp. 281-306.

[5] J. Chen, K Chen, J. Wu, and C.Chen, “A study of the flexible job shop scheduling problem with parallel machines and reentrant process,” Int J. Adv Manuf. Technol., 2008, 39(3*)*, pp. 344–354.

[6] M. Mastrolilli, L.M. Gamberdella, “Effective neighborhood for the flexible job shop problem,” J. of Scheduling, 2000, Vol. 3, no.1, pp. 3-20.

[7] D.Y. Sha, Lin Hsing-Hung, “A multi-objective PSO for job-shop scheduling problems,” Expert Systems with Applications 37, 2010, pp. 1065–1070.

[8] Xia, Z.Wu, “An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems,” Computer Ind. Eng. 48(2), 2005, pp. 409–425.

[9] I. Kacem, Hammadi, and P. Borne, “Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic,” Math Comput. Simul. 60(3–5), 2002, pp. 245–276,

[10] J.Gao, L.Sun, and M.Gen, “A hybrid genetic and variable neighborhood descent algorithm for flexible job shop

scheduling problems," Comput. Oper. Res. 35(9), 2008, pp. 2892–2907.

[11] Li Jun-qing, and Yu-xia Pan, "A hybrid discrete particle swarm optimization algorithm for solving fuzzy job shop scheduling problem," Int. J. Adv Manuf Technol. 66, 2013, pp. 583–596.

[12] R.Tavakkoli-Moghaddam, M.Azarkish and A.Sadeghnejad-Barkousaraie, "A hybrid algorithm based on particle swarm optimization and simulated annealing for periodic job shop scheduling problem," Expert Systems with Applications 38, 20011, pp. 10812–10821.

[13] Zhang Guohui, Shao Xinyu, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem," Computers & Industrial Engineering 56, 2009, pp.1309–1318.

[14] P.Fattahi, M. Manesh, and A. Roshani, "A new solution seed for job shop scheduling problem," Appl. Mech. Mater 110–116, 2012, pp. 3899–3905.

[15] R.Carlo, C.Raquel, Jr. Prospero Naval, "An Effective Use of Crowding Distance in Multi-objective Particle Swarm Optimization," IEEE Processding,2008

[16] D.Lei, "A Pareto archive particle swarm optimization for multi-objective job shop scheduling," Comput Ind. Eng. 54(4), 2008, pp. 960–971.

[17] A.Bagheri, Zandiesh, "An artificial immune algorithm for the flexible job shop scheduling problem," Futur Gener. Compter Sys 26(4), 2010, pp. 533-541.

[18] Shao Xinyu, Weiqi Liu, Liu Qiong, Zhang Chaoyong, "Hybrid discrete particle swarm optimization for multi-objective flexible job shop scheduling problem," Int. J. Adv Manuf Tech., 2013, pp. 2885–2901.

[19] G.Zhang, X. Shao, P.Li, and Gao, "An effective hybrid particle swarm optimization algorithm for multi-objective flexible job-shop scheduling problem," Computer Ind. Eng 56(4), 2009, pp. 1309–1318.

[20] J.Kennedy, R. Eberhart, "Particle swarm optimization," Proc IEEE Int. Conf. Neural Network 194, 1995, pp 1942–1948.

[21] Wang Hui-Mei, Fuh-Der Chou, and Ful-Chiang Wu, "A simulated annealing for hybrid flow shop scheduling with multiprocessor tasks to minimize makespan," Int. J. Adv. Manuf Technol., 2011, pp. 761–776.

[22] Li. Xing Ning , Wu Ying, Chen, Ke-Wei Yang, " An efficient search method for multi-objective flexible job shop scheduling problems," J. Intell Manuf, 2009, pp. 283–293.

[23] H.Chen, J. Ihlow, C. Lehmann, "A genetic algorithm for flexible job shop scheduling problem," In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1120–1125.

[24] Zhang Liping, Gao Liang, Li Xinku, "A hybrid intelligent algorithm and rescheduling technique for job shop scheduling problem with disruptions," Int. J. Adv. Manfuc. Tech., 2013, pp. 1141-1156.

[25] H.Wei ,D. Dun, " Scheduling flexible job shop problem subject to machine breakdown with route changing and right shift strategies," Int. J. Adv. Manf. Tech., 2013, pp. 501-514.

[26] P.R.Dian and M.S.Siti, "Particle Swarm Optimization: Technique, System & challenges," Int. J. Comp. App. (0975-8887), 2011.

[27] L.N.Xing, Y.Chen, K.W. Yang, "An efficient search method for multi-objective flexible job shop scheduling problems," J. Intell. Manuf., 2009, pp: 283-293. 2013.

[28] Sudip Kumar Sahana, Aruna Jain, Prabhat Kumar Mahanti, "Ant Colony Optimization for Train Scheduling: An Analysis," Int.J. Intelligent Systems and Applications, 2014, 02, 29-36.

[29] Mustapha Guezouri, Abdelkrim Houacin, " Hybrid Flow Shop Scheduling Problem Using Artificial Immune System," Int. J. Intelligent Systems and Applications, 2012, 10, 82-88.

Authors' Profiles

S. V. Kamble is presently working as an Assistant Professor in Information Technology at DKTE's Textile and Engineering Institute, Ichalkaranji, MS, India. He has received his Bachelor of Engineering (BE) in Information Technology from TKIET, Warananagar, MS, India and his Master of Technology (M.Tech.) in Computer Science and Engineering (CSE) from RIT, Sakhrale, MS, India.

He has 9 years of teaching experience. His research interests include Scheduling and Optimizations. He has published about 06 research papers in Conferences and Journals.

S. U. Mane is presently working as an Assistant Professor in Computer Science and Engineering at RIT, Sakhrale, MS, India. He has received his Bachelor of Engineering (BE) in Information Technology from BVCOE, Kolhapur, MS, India and his Master of Technology (M.Tech.) in Computer Science and Engineering (CSE) from BATU, Lonere, MS, India.

He has 7 years of teaching experience at UG and 3 years at PG. His research interests include High Performance Computing using GPGPU, Searching, Scheduling and Optimization Methods, Parallelization of Heuristic and Meta-Heuristic Methods, Constraint Satisfaction Problems and Scheduling Problems. He has published about 8 research papers in Conferences and Journals.

A. J. Umbarkar is presently working as an Assistant Professor in Information Technology at Walchand College of Engineering, at Sangli, MS, India. He has received his Bachelor of Engineering (BE) in Computer Engineering from PICT, Pune, MS, India and his Master of Engineering (ME) from Computer Science and Engineering (CSE) from Walchand College of Engineering, Sangli, MS, India.

He has 14 years of teaching experience at UG and 8 years at PG. His research interests include Parallel Genetic Algorithms, Parallel Evolutionary Algorithms and Parallel programming. He has published about 18 research papers in Conferences and Journals.

How to cite this paper: S. V. Kamble, S. U. Mane, A. J. Umbarkar,"Hybrid Multi-Objective Particle Swarm Optimization for Flexible Job Shop Scheduling Problem", International Journal of Intelligent Systems and Applications (IJISA), vol.7, no.4, pp.54-61, 2015. DOI: 10.5815/ijisa.2015.04.08