IJEE 0973-7383 Volume-9 • Number-2 June -Dec 2017 pp. 16-21 available online at www.csjournalss.com

Particle Swarm Optimization with Mutation for Flexible Job Shop Scheduling Problem

Dr. Meenu Dave, Kirti Choudhary
Department of Computer Science, Jagannath University Jaipur, India

Abstract: The flexible job shop scheduling is an extension of a job shop scheduling problem. FJSP implies that any task is performed on a machine from a given set of machines. Primarily it's employed in versatile producing system (FMS). Because the range of jobs increase, it becomes harder to get the optimum schedule in a very given amount of time. This paper considers the flexible flow shops scheduling problem. The target is to attenuate the make span. In this paper, AN improved particle swarm optimization rule for optimization of the flexible job-shop scheduling problem with mutation operator is given, that is employed to introduce diversity within the search procedure. Once the amendment of the total archive tends to decrease, the mutation method can begin.
Keywords: Flexible Job Shop Scheduling Problem, Scheduling, Mutation Operator.

I. INTRODUCTION

The FJSP is taken into account as associate extension of the normal job shop scheduling problem with further constraint of associate operation of employment is processed in additional than one facility. The versatile job shop problem is to [1,2] organize the execution of n jobs on m machines. during this problem, there are unit a collection of machines, k=1,2,…,m , and a collection of jobs, i=1,2,…,n so every job consists of a preset sequence of operations.

The objective of the matter is to assign every operation to associate applicable machine and sequence the operations on the machines so as to reduce the makespan that is that the time needed finishing all the roles. To [3] Minimize the makespan follow these strategy-

1. Minimize the machine's unproductive time
2. End every job within given time
3. Minimize the in method inventory prices

Particle swarm optimisation usually [4] converges comparatively speedily at the start of the search and so stagnates because of loss of diversity within the population. To beat this disadvantage, mutation, a wide used operator in genetic algorithmic rule, is employed to introduce diversity within the search procedure. Once the modification of the full archive tends to decrease, the mutation method [5] can begin. If variety the amount the quantity of iteration is a smaller amount than the merchandise of most number of iteration and chance of mutation then solely the mutation is performed on the position of the particle.

II. PROBLEM REPRESENTATION

i) Problem representation of FJSP-
In this work, [6] a true variety encoding system is projected. The whole number part is employed to assign the operations of every job to the machine and fractional half is employed to sequence of the operations on each machine. The position of the every particle is delineating by a true variety. The worth [7] of whole number half portion as a priority level for every operation that is employed to pick out the machine for the operation. 1st sequencing of obtainable machines for AN operation consistent with the increasing order of time interval is dole out. If tie happens, the machine having lower variety is given the

priority. Priority levels for all machines are generated for process all the operations of every job [8]. As AN instance, a drag is to execute 3 jobs on four machines. Table one represents knowledge together with jobs, operations, and process times on totally different machines. Table 2 shows the order of priority or priority level i.e. 1, 2, 3, four of machines adore every operation.

Table 1 Example problem

Jobs	Operations	M 1	M 2	M 3	M 4
J 1	O1,1	2	3	4	8
	O1,2	4	8	7	6
	O1,3	2	7	7	4
J 2	O2,1	7	4	5	3
	O2,2	1	5	3	4
J 3	O3,1	2	5	7	2
	O3,2	1	1	4	4

Table 2 Priority order

Jobs	Operations	P 1	P 2	P 3	P 4
J 1	O1,1	M1	M2	M3	M4
	O1,2	M3	M2	M1	M4
	O1,3	M3	M2	M1	M4
J 2	O2,1	M4	M2	M3	M1
	O2,2	M3	M1	M2	M4
J 3	O3,1	M2	M3	M4	M1
	O3,2	M2	M1	M4	M3

Table 3 represents the stochastic particle position representation. Initial particle positions in the swarm are generated by random number distributed uniformly on [xmin, xmax] where xmin = 1.0, xmax = mpl. The maximum positionxmax of the particle is taken as the maximum value of priority level (mpl) i.e. the number of machines available. The position of the particle must be a positive integer as each particle position value represents priority level for each operation. Hence, it lies in the range [1, mpl].

For example, the 1st position is 2.25 and the integer value is 2. Therefore, operation O1, 1 is assigned to machine 3 as per the priority order in Table 2. The process order of operations to be scheduled on the same machine depends on the value of fractional parts. The operations are sequenced according to the ascending order of the fractional part which is processed by the same machine. For instance, operations O1,2 and O3,2 are assigned to machine 2. The sequence of operations to be scheduled on machine 2 is operation (O3,2) followed by operation(O1,2) because the fractional part of the particle position for O3,2 is greater than fractional part of the particle position for O1,2. If the value of fractional parts is equal then the operation processing sequence is randomly chosen.

Table 3 A stochastic particle position representation

Operation	O1,1	O1,2	O1,3	O2,1	O2,2	O3,1	O3,2
Positions	3.35	4.62`	2.16	3.21	1.24	4.23	2.22
priority	3	4	2	3	1	4	2
Processing machine	M3	M4	M2	M3	M1	M4	M2

i. problem representation of Mutation operator-

Given a particle, a every which way chosen variable, mp, is mutated to assume a worth m'p as given by following equation.

$$m'_p = \begin{cases} m_p + \Delta(t, UB - m_p) & \text{if flip} = 0 \\ m_p - \Delta(t, m_p - LB) & \text{if flip} = 1 \end{cases}$$

When flip denotes the random event of returning {0 or one. UB and lb denote the higher and bound of the variable m_prespectively. The [9] perform Δ(t,x) returns a worth within the vary [0,x] specified the chance of Δ(t,x) being near zero will increase as t will increase.

$$\Delta(t, x) = x \times \left(1 - r^{\left(1 - \frac{t}{MAXT}\right)^b}\right)$$

Where r is that the random variety generated within the vary [0, 1], MAXT is that the most variety of iterations and t is that the variety of iteration. The parameter b determines the degree of dependence of mutation on the iteration variety.

I. PROPOSED PSO WITH MUTATION

The projected methodology introduces a hybrid algorithm by combining PSO with chaotic numbers and mutation operator is embedded to flee from native optima and to enhance the answer diversity. varied steps concerned within the projected PSO algorithm are listed out below.

Step 1. Initialize the parameters such as population size, maximum iteration, decrement factor, inertia weight, social and cognitive parameters.

Step 2. Input numbers of jobs, and number of machines at each stage and processing times.

Step 3. Generate the initial position and velocities values of the particle of the particle.

Step 4 . Get the schedule using encoding scheme.

Step 5. Evaluate each particle's fitness(makespan).

Step 6 . Find out the personal best(Pbest) and global best(Gbest).

Step 7 . If (t<(tmax * PMUT), then perform mutation on Xtij.

(PMUT is the probability of mutation).

Step 8. Update velocity, position and inertia weight.

Step 9. Terminate if maximum number of iteration is reached and store the gbest.

II. RESULTS

The The procedure study aims to research the performance of PSO to reduce the makespan for the versatile job shop scheduling problems. The formula was enforced in Matlab ten on a Pentium IV running at 2 gigahertz on the Windows7 software system. The projected algorithm is tested on Brandimarte (1993)'s (BR) [10] data set that contains a collection of ten problems. the amount of jobs ranges from ten to twenty, range|the amount|the quantity} of machine ranges from four to fifteen and also the number of operations for every job ranges from five to fifteen. This knowledge set is that the most typically adopted benchmark instances within the literature on FJSP. In Table 4, comparison of makespan obtained by PSO to the results of the GA (Pezzella et al) [11], integrated genetic algorithm (Wu et al.) [12] and hybrid PSO(Jun li et al.) [13] on 10 FJSP instances from Brandimarte knowledge set is formed.

The first and second columns symbolize the name and size of the matter severally. The third column refers to the most effective makespan result from projected PSO formula. The remainder column up to the tip one represent the most effective makespan resulted from GA, integrated genetic formula and PSO severally. Relative deviation criterion is employed to check the results of the PSO with those of the on top of 5 mentioned algorithms.

Relative deviation is obtained as follows:

$$\mathrm{Dev}(\%) = \frac{(C_{max})_i - C_{max}}{(C_{max})_i} \times 100$$

Where $(C_{max})_i$ denotes the makespan of alternative formulas and Cmax denotes the makespan obtained by our projected PSO algorithm.

Table 4 Results of the BR data instances

Problem	**n x m**	**Proposed PSO**	**GA**		**Integrated GA**		**Hybrid PSO**	
			Cmax	**Dev(%)**	**Cmax**	**Dev(%)**	**Cmax**	**Dev(%)**
Mk 01	**10 x 6**	37	40	7.5	40	7.5	40	8.1
Mk 02	**10 x 6**	31	31	0	27	3.7	27	-12.9
Mk 03	**15 x 8**	204	204	0	204	0	204	0.0
Mk 04	**15 x 8**	60	60	0	60	0	63	5.0
Mk 05	**15 x 4**	173	173	0	173	0	173	0.0
Mk 06	**10x15**	64	63	-3.17	62	-4.84	65	1.6
Mk 07	**20 x 5**	139	139	0	139	0	145	4.3
Mk 08	**20x10**	523	523	0	523	0	523	0.0
Mk 09	**20x10**	307	311	1.29	309	0.65	331	7.8
Mk 10	**20x15**	205	212	3.3	206	0.49	223	8.8

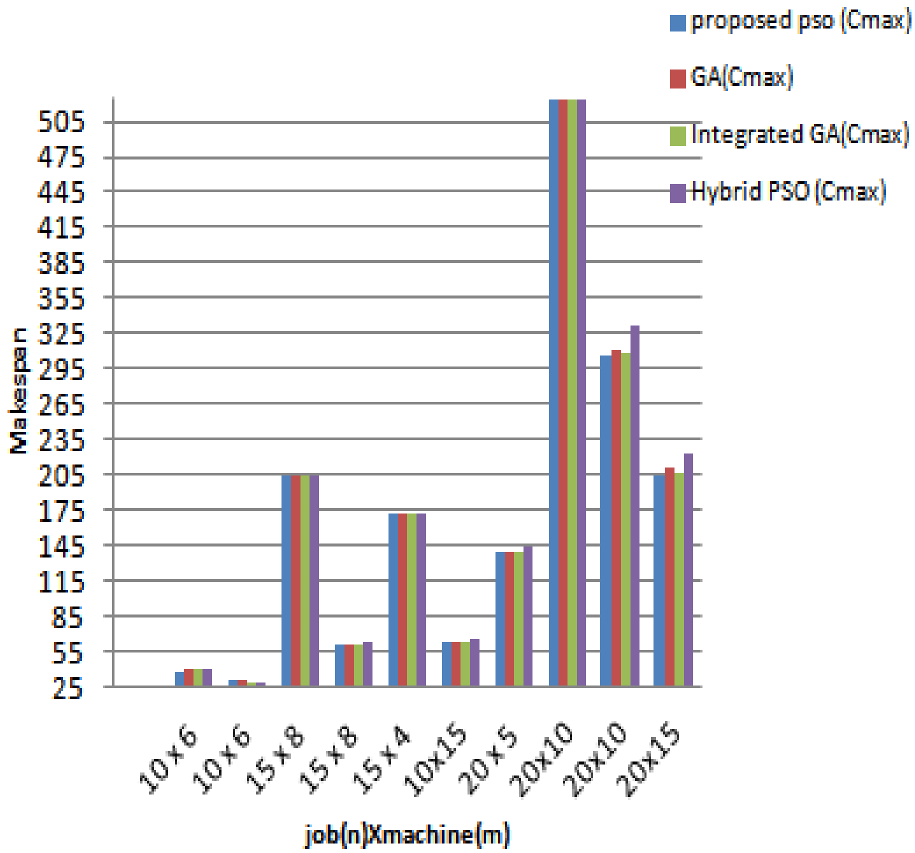

Fig. 1. Comparative makespan results

III. CONCLUSION

In this analysis, flexible job shop scheduling problem that may be a NP-hard problem is taken into account and an economical quantum particle swarm optimisation to seek out near-optimal schedules has been used. The mutation operator employed in genetic algorithmic rule is embedded in PSO to avoid premature convergence and improve answer diversity and reduces procedure burden. The planned PSO approach is found to be a decent problem finding technique for scheduling problem. The algorithm is applied on a set of problem instances from Kacem et al. [14], Brandimarte [10] . The results indicate that PSO produces either better solutions or same as compared to best known solutions in the literature. It has been demonstrated that PSO outperforms other well-known algorithms at least for benchmark instances considered in the study.

IV. FUTURE SCOPE

The study may be extended to dynamic stochastic shop scheduling where the jobs arrive continuously in time. More optimum schedule can be produced by considering other performance measures like work load on a critical machine, machine loading capacity, etc.

REFERENCES

[1] Campbell, H.G., Dudek, R.A., and Smith, M.L., "A heuristic algorithm for the n-job, m-machine sequencing problem", Management Science 16, B630-B637,(1970).

[2] Paolo Brandimarte. Routing and scheduling in a _exible job shop by tabu search. Annals of Operations Research, 41(3):157_183, 1993.

[3] Prabhaharan,S.G., Jawahar,S. hahul Hamid Khan,B., Asokan,P.,and Thiyagu M.,(2005). "A Particle swarm optimization algorithm for permutation flow shop scheduling with regular and non regular measures",International journal of applied management and technology,vol.3,no:1,pp 171-182.

[4] Singh, M. R., & Mahapatra, S. S. (2012). A swarm optimization approach for flexible flow shop scheduling with multiprocessor tasks. International Journal of Advanced Manufacturing Technology, 62(1–4), 267–277

[5] Tsung-Che Chiang n, Hsiao-JouLin , A simple and effective evolutionary algorithm for multiobjective flexible job shop scheduling , Int. J. Production Economics 141 (2013) 87–98

[6] Xing LN, Chen YW, Yang KW (2009) An efficient search method for multi-objective flexible job shop scheduling problems. J Intell Manuf 20:283–293

[7] Wang, X., Gao, L., Zhang, C., Shao, X., "A multi-objective genetic algorithm based on immune and entropy principle for flexible job-shop scheduling problem." International Journal of Advanced Manufacturing Technology 2010, 51, 757–767.

[8] Xia, W. J., & Wu, Z. M. (2005). An effective hybrid optimization approach for multiobjective flexible job-shop scheduling problems. Computers and Industrial Engineering, 48(2), 409–425.

[9] L. De Giovanni and F. Pezzella, "An improved genetic algorithm for the distributed and flexible job-shop scheduling problem", European Journal of Operational Research, vol. 200, no. 2, pp. 395-408, 2010.

[10] Brandimarte, P. (1993), Routing and scheduling in a flexible job shop by tabu search, Annals of Operations Research, 41: 157-183.

[11] Pezzella F, Morganti G, Ciaschetti G. A genetic algorithm for the flexible job-shop scheduling problem. Computer & Operation Research 2008; 35(10):3202-212.

[12] Wu W, Cai H, Jiang L. Integrated genetic algorithm for flexible job-shop scheduling problem. Computer Engineering and Applications 2009; 45 (22): 183-86.

[13] Jun-qing Li et al, " A hybrid particle swarm optimization and tabu search algorithm for flexible job-shop scheduling problem ",International Journal of Computer Theory and Engineering2010, Vol. 2, pp.1793-1801

[14] Kacem, I., Hammadi, S. and Borne, P, "Approach by localization and multi-objective evolutionary optimization for flexible job-shop scheduling problems", IEEE Transactions on Systems, Man and Cybernetics, Part C, 2002, 32(1), pp:408-419.

[15] Garey, M.R., Johnson, D.S., and Sethi, R., "Complexity of flowshop and jobshop scheduling", Mathematics of Operations Research 1 ,117-129. (1976).

[16] Page,E.S., "An approach to the scheduling of jobs on machines", journal of the royal statistical Society .series B 23 ,484-492. (1961).