IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE)
e- ISSN: 2278-1684, p-ISSN : 2320–334X
PP 01-05
www.iosrjournals.org

A COMPARISON OF GENETIC ALGORITHM AND PARTICLE SWARM OPTIMIZATION TO OPTIMIZE MULTI OBJECTIVE FOR FJSP

B.Santhosh Shanmuga Pandian1, Dr R.Kesavan2
1(Production Technology, PG Scholar, MIT Campus, Anna univ Chennai Imdii)
.2(Production Technology, Associate Professor, MIT Campus, Anna univ Chennai.indi).

ABSTRACT**:** *The job-shop scheduling is one of the useful techniques for solving optimization of problems. Flexible job-shop scheduling problem is an alternative tool of the job-shop problem that allows a process to be carried on by the machine from the given alternative ways. There are available many algorithms to solve the flexible job-shop scheduling problems. The aims are to minimize the overall completion time (makespan), the total workload of machines and the workload of the most loaded machine. It also minimizes the manufacturing cost. Genetic Algorithm and particle swarm optimization are implemented to solve the multi-objective flexible job-shop scheduling problem. Both are population based algorithm, we compare the performance of these algorithms to achieve an optimal or near optimal solution. It is concluded that the particle swarm optimization algorithm gives promising solutions. The excellence of solution achieved by particle swarm optimization algorithm is greater to that of the genetic algorithm, but the genetic algorithm takes shorter time to find a schedule solution.*

Keywords *Flexible job-shop scheduling; Multi-objective optimization; Non-dominated Sorting Genetic Algorithm; Particle Swarm Optimization*

1. INTRODUCTION

Here, *n* jobs are carried on to *m* unrelated machines. To be cope with the modern competitive market needs, manufacturing methods are to be more flexible [1]. Now, a machine has the strength of performing more than one type of operation. It results in a modified version called the flexible JSP [2], flexible job-shop scheduling problem (FJSP) is an extended traditional job-shop scheduling problem. It sweeps out the restriction of unique resources and permits each operation to be carried on by several different machines, thus makes it possible to schedule the problem to coincide with actual production situation. FJSP needs to allot each activity to a machine from a set of machines and then sequence the assigned operations on each machine, referring to the paper by Xia and Wu [5]. Brucker and Schlie [6] dealt will carried the problem that one operation could be understood on several machines and have studied this problem deeply as pioneer, and the beginning of the study on FJSP.Hierarch approached & integrated approach can be of good to solve this kind of problem only it is.The farmer approach, which was first proposed by Brandimarte [7], considered the assigning sub-problem and the sequencing sub-problem individually, to decompose the complex problem into some sub-problems in order to reduce the complexity. However, the integrated approaches solve the assigning sub-problem and the sequencing sub-problem simultaneously, such as greedy heuristics [8], simulated annealing (SA) algorithm [9] and tabu search [10].FJSP has been concentrated mostly on single objective. However, several objectives are being take care simultaneously in the real-world production situation. It has won attention of some researchers. Kacem et al. [11-13] used an approach by localization and multi-objective evolutionary optimization and proposed a Pareto approach based on the hybridization of fuzzy logic (FL) and evolutionary algorithms (EAs) to solve the FJSP. Baykasoğlu et al. [14] presented a linguistic-based meta-heuristic modeling and multiple objective tabu search algorithm to solve the flexible job-shop scheduling problem. Xia and Wu [5] proposed a practical hierarchical solution approach for solving MOFJSP.The proposed approach utilizes particle swarm optimization (PSO) to assign operations on machines and simulated annealing (SA) algorithm to schedule operations on each machine. Liu et al. [15] proposed the variable neighborhood particle swarm optimization including of a set of the variable neighborhood search and particle swarm optimization (PSO) for solving the multi-objective flexible job-shop scheduling problems.

2. MATHEMATICAL FORMULATION OF FJSP

It is considered that organizing denoted by Jjobs on M machines. The set of machines is V. Each job $J_j$denotes a number of $n_j$non-preemptable ordered operations (precedence constraint). The execution of the i_{th} operation of job J_j(noted $O_{i\ ,j}$) requires a resource or machine selected from a set of available machines. The assignment of the operation $O_{i,\ j}$to the machine M_k isthe occupation of this machine during a processing time called $d_{i,j,k}$. In this problem, all machines are available at t = 0 and each job $J_j$can be started at t = 0*; at* a given time, a machine can only execute one operation: it becomes available to other operations once the operation which is currently assigned to it is completed (resource constraints);

Following objectives are to be minimized:

Themakespan (G$_1$):$f_1 = \max(tf_{n_j,j})$ (1)

The total manufacturing cost (G$_2$):$f_2 = \sum_k M_c$ (2)

High speed machines are better than low speed machines and high speed machines take less processing time to produce a job compared to low speed machines. Hence, the high speed machines result in more manufacturing cost, and vice versa. The following function is used to evaluate the unit manufacturing cost.

Manufacturing cost of a machine per unit time$M_c = \frac{A\times e^{-k\times\sum P_T}}{\sum P_T}$ (3)

Here, the constant coefficient $A$is fixed costs is with the value of 1000. The term e is exponential function. The term $k$fixes the cost of producing a single job to the required dimension and includes the charge rate of the machine. $\sum P_T$is total processing time of all jobs on a machine.

3. PROPOSED GA FOR FJSP

In this work, GA is taken to optimize the objectives in the flexible job-shop problems. The test samples (Kacem instances) for this study are seen in Kacem et al. [9]. The processing time of the instances 4 jobs × 5 machines is shown in Table 1. Thus production cost is calculated for each machine, using exponential function.

3.1 INPUT MODULE

1. Number of jobs = 4
2. Number of machines = 5
3. Processing times given in Table 1
4. Manufacturing cost given in Table 1

3.2 INITIALIZATION MODULE

Randomly formed ten chromosomes, it is the initial population. The pheno style coding is handled to represent the solutions of the problem as chromosomes. Every chromosome has two substrings. Each substring consists of 12 bits, the genes of the chromosomes. Take the first chromosome, it is {431213525323 | 3124}. The first substring, {431213525323}, represents the machine assignment, and the second substring {3124} represents schedule of job sequence.

Table 1 Processing time and manufacturing cost for 4 jobs × 5 machines problem

Manufacturing cost ($/unit time)	M_1	M_2	M_3	M_4	M_5
$O_{i,j}$	\$15.97	\$13.81	\$15.97	\$5.98	\$10.89
J_1					
$O_{1,1}$	2	5	4	1	2
$O_{2,1}$	5	4	5	7	5

$O_{3,1}$	4	5	5	4	5
J_2					
$O_{1,2}$	2	5	4	7	8
$O_{2,2}$	5	6	9	8	5
$O_{3,2}$	4	5	4	54	5
J_3					
$O_{1,3}$	9	8	6	7	9
$O_{2,3}$	6	1	2	5	4
$O_{3,3}$	2	5	4	2	4
$O_{4,3}$	4	5	2	1	5
J_4					
$O_{1,4}$	1	5	2	4	12
$O_{2,4}$	5	1	2	1	2

3.3EVALUATION MODULE

The make span, the workload of the most loaded machine, the total workload of machines are to be reduced, that is the objective. Where, the total manufacturing cost is the flexible. Equations (1), (2), (3) and (4) the objective values, are taken to have four objectives *fit* (1) and *fit* (4) are calculated.

3.4 FITNESS FUNCTION

New fit (X) = $(1/Z_1)$ x K where fit(X) is the fitness parameter value and k= 10 is a constant applied to scale the fitness function

3.5 TERMINATION CRITERION

It the iteration number is at its maximum value, the GA is excluded. After a few pilot runs, it is taken as the maximum number of iteration

4. PROPOSED PSO FOR FJSP

In this work, PSO is taken to optimize the objectives in the flexible job-shop problems. The test samples (Kacem instances) for this study are seen in Kacem et al. [9]. The processing time of the instances 4 jobs × 5 machines is shown in Table 1.PSO is a population-based optimization technique inspired by the choreography of a bird flock. This technique has virtuous performance, low computational cost and relaxed implementation. A set particle dimension as equal to the size of ready tasks, Initialize particles position randomly for each particle, calculate its fitness value. If the fitness value is better than the previous best pbest, set the current fitness value as the new pbest.After Steps 3 and 4 for all particles, select the best particle as gbest.For all particles, calculate velocity using Equation 1 and update their positions using Equation 2.If the stopping criteria or maximum iteration is not satisfied, repeat from Step3.

4.1 INPUT MODULE

The input data required is

1. Number of jobs = 4
2. Number of machines = 5
3. Processing times given in Table 1
4. Manufacturing cost given in Table 1

4.2 INITIALIZATION MODULE

Five possible combinations of tasks are considered in the initialization module. Consists of a five particles. Each and every particle in the initial population has a single row.

4.3 EVALUATION MODULE

Initial module matrixes called s_k^i and the manufacturing system efficiency is calculated for all particles. The maximum total indexing time in seconds is corresponding to the second particle hence; the second particle is selected and is called $g_{best}p_k^g$ for the first iteration. The p_k^g is then converted into the same size of s_k^i, by repeating the same row. The particle will not move at the beginning. So, for the first iteration, the initial velocity v_k^i is taken as zero and $v_k^i=s_k^i$. rand () is a matrix created by random numbers between 0 and 1.Inertia weight, ω is decreased linearly from 0.9 to 0.4. C_1 and C_2 are learning factors: $C_a = C_b = 2$. Vmax, the maximum change that one particle can take during one iteration is taken as 5.

Substituting the above values in the equation,

$$v_{k+1}^{i} = \omega * v_k^i + c_a * rand() * \left(p_k^i - s_k^i \right) + c_b * rand() * \left(p_k^g - s_k^i \right)$$

$$= c_b * rand() * \left(p_k^g - s_k^i \right)$$

Firstly, to find, initial module matrix is subtracted from global best matrix, generating random numbers from 0to 1.Global best Matrix and random matrix are multiplied. Where Cb(C_b=2). Modulus of 5 is taken for first 12 cell value of this resultant matrix, so that all the entries in the matrix will be less than 5.v_{k+1}^iFor the first iteration. Is obtained by adding s_k^i andv_{k+1}^i. Modulus of 4 is taken for remaining cell value of this resultant matrix. The repetition of value is overcome by altering the repeating value of the missing cell value showss_{k+1}^i.From the second iteration onwards, s_{k+1}^i becomes s_k^i , and v_{k+1}^ibecomes v_k^i. The above procedure is repeated until the termination criterion is met.

4.4 TERMINATION CRITERION

The whole process of PSO is repeated for maximum number of consecutive iterations.

4.5 OUTPUT MODULE

The best of given task is obtained by PSO is given as the output.

Table2 Comparison of results

Problem	Objective	GA	PSO
		Solution	Solution
4 jobs × 5 machines	$G_{1\ (Min)}$	12	11
	$G_{2(Rs)}$	392.56	390.76

5. RESULT AND DISCUSSION

All these are coded in Visual studio C language and the experiments were done on an Intel Process computer. It is recognised that this algorithm outperforms other known GA PSO for the same problem, and gives results comparable with the best algorithm known so far. The proposed algorithm is tested on benchmark problems found in Kacem et al. [12]. Pareto optimal solutions are obtained for the four instances, i.e., 4 jobs × 5 machines problem. The best solutions are shown in Table 2. The PSO algorithm is more efficient than Genetic algorithms.

6. CONCLUSION

This study outing the problems related to FJSP, which aims at reducing the make span, and the total manufacturing cost, as a nonlinear programming and multi objective problem. The findings are applied with the well-known instances of flexible job-shop problem,it is observed that PSO gets the optimal solution in quicker time.As introduction of some other hybrid algorithms to solve the multiobjective FJSP will be an interesting subject. This study also considers some factors like transportation time, traffic regulation, etc. which can enrich the proposed approach and give scientific benefits.

REFERENCES

[1]. Saidi-Mehrabad M, Fattahi P (2007) Flexible job shop scheduling with tabu search

[2]. Chen JC, Chen KH, WU JJ, Chen CW (2008)a study of the flexible job shop scheduling problem with parallel machines and reentrant process. Int J AdvManuftechnol 39(3-4):344-354

[3]. Li JQ, Pan QK, Suganthan, PN and Chua TJ (2011) A hybrid tabu search algorithm with an efficient neighborhood structure for the flexible job shop scheduling problem. Int J AdvManufTechnol 52:683-697

[4]. Wang X, Gao L, Zhang C and Shao X (2010) A multi-objective genetic algorithm based on immune and entropy principle for flexible job shop scheduling problem. Int J AdvManufTechnol 51:757-767

[5]. Xia WJ, Wu ZM (2005) An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. ComputIndEng 48(2):409–425

[6]. Brucker P, Schlie R (1990) Job-shop scheduling with multipurpose machines. Computing 45(4):369–375

[7]. Brandimarte P (1993) Routing and scheduling in a flexible job shop by taboo search. Ann Oper Res 41:157–183

[8]. Mati Y, RezgN,Xie XL (2001) An integrated greedy heuristic for a flexible job shop scheduling problem. Proceedings of the 2001 IEEE International Conference on Systems, Man, and Cybernetics. OPAC, Tucson, pp 2534–2539

[9]. Hapke M, Jaszkiewicz A, Słowiński R (2000) Pareto simulated annealing for fuzzy multi-objective combinatorial optimization. J Heuristics 6(3):329–345

[10]. Scrich CR, Armentano VA, Laguna M (2004) Tardiness minimization in a flexible job shop: a tabu search approach. J Intell Manuf 15(1):103–115

[11]. Kacem I, Hammadi S, Borne P (2002a) Pareto-optimality approach for flexible job-shop scheduling problems: hybridization of evolutionary algorithms and fuzzy logic. Math ComputSimul 60 (3–5):245–276

[12]. Kacem I, Hammadi S, Borne P (2002b) Approach by localization and multiobjective evolutionary optimization for flexible job-shop scheduling problems. IEEE Sys Man Cybern 32(2):172–172

[13]. Kacem I, Hammadi S, Borne P (2002c) Approach by localization and multiobjectiveevolutionary optimization for flexible job-shop scheduling problems. IEEE Sys Man Cybern 32(1):1–13

[14]. 15. Baykasoğlu A, Özbakir L, Sönmez A (2004) Using multiple objective tabu search and grammars to model and solve multiobjective flexible job shop scheduling problems. J Intell Manuf15 (6):777–785