ISSN 2321-807X
Volume 13 Number7
Journal of Advances in chemistry

AIS-SFHM APPROACH FOR OPTIMIZATION OF MULTI OBJECTIVE JOB SHOP PROBLEMS

S.Gobinath[1], M.Chandrasekaran[2], P.Sriramya[3]
[1]Professor, Dept of Mechanical Engineering, Kongunadu College of Engineering and Technology, Thottium, Trichy Dist, Tamilnadu, India
[2,] Professor, Department of Mechanical Engineering, Vels University, Chennai, India.
[3]Saveetha University, Chennai, India
*nithnathdeep@yahoo.com

Abstract

The n-job, m-machine Job shop scheduling (JSP) problem is one of the general production scheduling problems. The JSP problem is a scheduling problem, where a set of 'n' jobs must be processed or assembled on a set of 'm' dedicated machines. Each job consists of a specific set of operations, which have to be processed according to a given technical precedence order. Job shop scheduling problem is a NP-hard combinatorial optimization problem. In this paper, optimization of three practical performance measures mean job flow time, mean job tardiness and makespan are considered. The hybrid approach of Sheep Flocks Heredity Model Algorithm (SFHM) is used for finding optimal makespan, mean flow time, mean tardiness. The hybrid SFHM approach is tested with multi objective job shop scheduling problems. Initial sequences are generated with Artificial Immune System (AIS) algorithm and results are refined using SFHM algorithm. The results show that the hybrid SFHM algorithm is an efficient and effective algorithm that gives better results than SFHM Algorithm, Genetic Algorithm (GA). The proposed hybrid SFHM algorithm is a good problem-solving technique for job shop scheduling problem with multi criteria.

Keywords: Multi objectives, Job shop scheduling, Sheep Flocks Heredity Model Algorithm, Artificial Immune System.

1.0 INTRODUCTION

The classical job-shop scheduling problem (JSP) is one of most difficult combinatorial optimization problems. During the last decades a great deal of attention has been paid to solving these problems with many kind of algorithms by considering single objective. But real world scheduling problems naturally involve multiple objectives. There are only few attempts available to tackle the multi-objective JSP.

In a multi-objective context, find as much different schedules as possible, which are non-dominated with regard to two or more objectives. Some frequently used performance measures are makespan, mean flow-time and mean tardiness. Makespan is defined as the maximum completion time of all jobs. Mean flow-time is the average of the flow-times of all jobs. Mean tardiness is defined as the average of tardiness of all jobs.

2.0 LITERATURE REVIEW

Bruker [1] show that the Job shop Scheduling (JSP) is an NP-hard [2] combinatorial problem. Because of the NP-hard characteristics of job shop scheduling, it is usually very hard to find its optimal solution, and an optimal solution in the mathematical sense is not always necessary in practices [3]. Researchers turned to search its near-optimal solutions with all kind of heuristic algorithms [4]. Fortunately, the searched near optimal solutions usually meet requirements of practical problems very well.

In a single-objective context some of the recent approaches have shown quite promising results [5-6]. But real world scheduling problems naturally involve multiple objectives. There are only few attempts to tackle the multi-objective JSP [7].

Additionally, researches on job shop scheduling problems have been concentrated primarily on the optimisation of individual measures of system performance. While a single objective may be justified in certain situations, many scheduling problems are more naturally formulated with multiple, often competing, objectives to obtain a trade-off schedule. Examples of multi-criteria scheduling approaches include those in Daniels [8], Lee and Jung [9], and Murata, Ishibuchi, and Tanaka [10].

The remainder of the paper is organized as follows. Section 3 describes the Problem statement. Section 4 introduces the new hybrid algorithm for JSP problem. Section 5 shows Results and discussion. Finally, section 6 presents the conclusion of this work.

3.0 PROBLEM STAREMENT

In a multi-objective context, find as much different schedules as possible, which are non-dominated with regard to two or more objectives. Performance measures are makespan, mean flow-time and mean tardiness. Makespan is defined as the maximum completion time of all jobs. Mean flow-time is the average of the flow-times of all jobs. Mean tardiness is defined as the average of tardiness of all jobs.

The combined objective function for the multi objective Job Shop Problem is,

COF=Min [w1 (msi/ms*) + w2 (Ti/T*) +
w3 (mfi/mf*)]

Where, w1 = (R1/∑R), w2 = (R2/∑R),
w3 = (R3/∑R)

∑R = (R1+ R2 +R3),

where R1, R2, R3 - Random numbers

ms*- Make Span Global minimum

T* - Mean Tardiness Global minimum

mf*- Mean Flow Time Global minimum

ms_i - Make span Iteration minimum

Ti - Mean Tardiness Iteration minimum

mfi- Mean Flow Time Iteration minimum

w1, w2, w3- Weightage factors

MFT- Mean flow time,

MT- Mean Tardiness

COF- Combined Objective Function

4.0 IMPLEMENTATION OF HYBRID ALGORITHM

4.1 Representation of solution seed

Consider the three-job three-machine problem

Processing time				Machine sequence			
Job	1	2	3	Job	1	2	3
J1	3	3	2	j1	m1	m2	m3
j2	1	5	3	j2	m1	m3	m2
j3	3	2	3	j3	m2	m1	m3

Suppose a seed is given as [3 2 1], where 1 stands for job j1, 2 for job j2, and 3 for job j3. This sequence has to be operated 3 times in the same order because each job has three operations. So that we can consider the initial seed as the following format [3 2 1 3 2 1 3 2 1]. There are three 2s in the seed, which stands for the three operations of job j2. The first 2 corresponds to the first operation of job j2 which will be processed on machine 1, the second 2 corresponds to the second operation of job j2 which will be processed on machine 3, and the third 2 corresponds to the third operation of job j2 which will be processed on machine 2. We can see that all operations for job j2 are given the same symbol 2 and then interpreted according to their orders of occurrence in the sequence of this seed. The corresponding relationships of the operations of jobs and processing machines are shown in Figure 1.

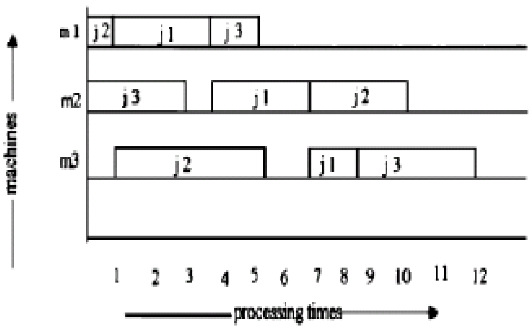

Figure.1. Feasible schedule

This concept is used to find the makespan for the sequences of the problems where the generated seed (job sequence) is operated equal to the number of machines represented in the particular problem.

4.2 Proposed hybrid Algorithm for Multi Objective JSP Problem

Generate a population of **P** *antibodies* (job sequences)

Stage 1(AIS Algorithm)

For each iteration

Select the sequence in the antibody population;

Find out the affinity of each antibody;

Cloning process (generate copies of the antibodies)

Steps in Mutation process (for each clone)

Find inverse mutation

Select the new sequence obtained from inverse mutation

Find the makespan of the new sequence

if (makespan *(*new sequence) = = makespan (clone))

then if (tardiness(new sequence) < tardiness (clone))

clone = new sequence ;

else clone = clone;

goto

if makespan *(*new sequence) < makespan (clone) then

Clone = *new* sequence

else,

do pair wise interchange

select the new sequence

Find the makespan of the new sequence

if (makespan *(*new sequence) = = makespan (clone))

then if (tardiness(new sequence) < tardiness (clone))

clone = new sequence

else clone = clone

goto

If makespan (new sequence) < makespan (clone) then

clone = new sequence:

else

clone = clone

antibody = clone

Eliminate worst %B number of antibodies in the population

Create new antibodies at the same number (%B of pop.)

change the eliminated ones with the new created ones

while stopping criteria = false.

Stage 2: (SFHM Algorithm)

Select the population,

Select the parent

Sub chromosome level crossover

Set sub chromosome level crossover probability

If population probability is less than or equal to sub chromosome level probability

Perform sub chromosome level crossover

Else retain the old sequences

Sub chromosome level mutation

Set sub chromosome mutation probability

If population probability is less than or equal to sub chromosome mutation probability

Perform sub chromosome level mutation

Else retain the same sequences

Select two sequences from population

Chromosome level crossover

Set crossover probability

If population probability is less than or equal to crossover probability

Perform chromosome level crossover

Else retain the same sequences

Chromosome level mutation

Set mutation probability

If population probability is less than or equal to mutation probability

Perform chromosome level mutation

Else retain the same sequences

End if terminal condition satisfied

5.0 RESULTS AND DISCUSSION

The Hybrid algorithm is implemented in C language on personal computer Pentium IV 2.4 GHz. The maximum number of iterations has been set to 100 X n, where n is the number of jobs.

Multi-objective optimization differs from single-objective optimization in many ways [11]. For two or more conflicting objectives, each objective corresponds to a different optimal solution, but none of these trade-off solutions is optimal with respect to all objectives. Thus, multi-objective optimization does not try to find one optimal solution but all trade-off solutions.

For multi-objective scheduling the proposed artificial immune algorithm is used to optimize makespan, mean flow time and mean tardiness of the two JSP given by Bagchi [7] are the basis of the following experiments. The first problem, called JSP1, is a ten job five machine instance. The second problem, called JSP2, is a ten job ten machine instance. Apparently, the hybrid algorithm minimizes all objectives simultaneously. This algorithm is compared with the similar previous work using GA [12] and SFHM algorithm [13] and shown in Table 1 and Table 2. Graph 2-9 shows that the comparison of makespan, mean flow time, mean tardiness and coefficient of function results of JSP1 and JSP2 problems.

6.0 CONCLUSION

In this paper, hybrid approach has been used for solving multi objective job shop scheduling problems with the objective of minimization of makespan, mean flow time and mean tardiness. The algorithm uses simple but effective techniques for calculating cloning process, applying mutations, a receptor editing procedure and multi stage genetic operation. This algorithm has been tested on JSP 1 and JSP 2 problem instances given in Bagchi [7]. The findings were compared with Genetic Algorithm [12] and SFHM algorithm [13] that tested the same problems. Hybrid algorithm gives better results than the genetic algorithm and SFHM algorithm. The proposed hybrid algorithm is competent and proves to be a good problem-solving technique for job shop scheduling.

7.0 REFERENCES

1. Bruker P, Scheduling Algorithms 2nd Edn, Springer-Verlag, Berlin, 1995.

2. Garey M et al, The complexity of flow shop and job shop scheduling, Mathematics of Operations Research, 1976, 1, pp 117-129.

3. Erschler JF, Roubellat JP, Vernhes, Finding some essential characteristics of the feasible solutions for a scheduling problem. Operations Research, , 1976, 24, pp 774-783.

4. French S, Sequencing and scheduling: An introduction to the mathematics of the job shop, New York, Wiley, 1982.

5. Mattfeld D.C, Evolutionary Search and the Job Shop, Physica-Verlag, 1996.

6. Ono I, Yamamura M, and Kobayashi S, A genetic algorithm for job-shop scheduling problems using job-based order crossover. In Proceedings of ICEC '96, 1996, pp 547-552.

7. Bagchi T.P, Multiobjective Scheduling By Genetic Algorithms, Kluwer Academic Publishers, 1999.

8. Daniels DL, Incorporating performance information into multiobjective scheduling. European Journal Operational Research, 1994, 77, pp 272–286.

9. Lee SM, Jung HJ, A multi-objective production planning model in A flexible manufacturing environment. International Journal Production Research, 1989, 27(11), pp 1981–1992.

10. Murata T, Hisao I, Tanaka H, Multi-objective genetic algorithm and its applications to flow shop scheduling. Computer Industrial Engineering, 1996, 30(6), pp 957–968.

11. Deb K., Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, 2001.

12. Garen J, Multi objective Job-Shop Scheduling with Genetic Algorithms Using a New Representation and Standard Uniform Crossover, MH WORKSHOP, 2003.

13. Chandrasekaran M et al., Multi Objective Optimization for Job shop scheduling using Sheep Flocks Heredity Model Algorithm, International journal of Manufacturing Science and Technology (In Review), 2006.

APPENDIX

Table 1. Results of Hybrid Algorithm with GA and SFHM algorithm for JSP1

JSP 1	Genetic Algorithm				Hybrid SFHM Algorithm				SFHM Algorithm			
s.no	Make span	MT	MFT	COF	Make span	MT	MFT	COF	Make span	MT	MFT	COF
1	156	10.8	128.4	0.92	109	6.530	94	0.54	141	11.3	124.5	0.56
2	158	8.2	126	0.90	128	6.542	100.7	0.56	142	11.3	120.8	0.57
3	159	15.7	124.3	0.62	131	6.395	105.8	0.42	148	11.5	117.5	0.45
4	159	7.8	127.3	0.77	138	6.534	69.5	0.49	149	11.2	120.5	0.50
5	160	13.9	124.3	0.63	139	6.534	88.2	0.48	150	11.2	123.0	0.50
6	162	6.4	130.5	0.67	145	6.534	101.9	0.46	152	11.3	128.6	0.49
7	165	6.4	128.8	0.86	147	6.389	88.1	0.51	153	11.3	125.1	0.53
8	167	15.1	122.4	0.62	149	5.449	78.5	0.51	156	11.2	116.9	0.52
9	169	6.1	134.5	0.68	150	6.412	95.5	0.44	161	11.3	109.6	0.45
10	182	5.8	135.4	0.63	152	6.396	112.2	0.46	167	11.1	112.3	0.47

Table 2. Results of Hybrid Algorithm with GA and SFHM algorithm for JSP2

JSP 2	Genetic Algorithm				Hybrid SFHM Algorithm				SFHM Algorithm			
s.no	Make span	MT	MFT	COF	Make span	MT	MFT	COF	Make span	MT	MFT	COF
1	196	32.2	174.7	0.72	186	28.7	161.5	0.60	188	29.9	163.6	0.62
2	199	33	174.6	0.70	189	27	157	0.29	190	29	158	0.31
3	201	31.8	176.1	0.62	203	25.3	156.9	0.36	205	28.7	157.1	0.37
4	203	32.2	173.4	0.67	205	26.9	155	0.39	206	28.4	158.6	0.40
5	204	31.3	174.	0.63	210	27	159	0.40	211	27	160	0.42

			8									
6	212	31.6	174.5	0.60	213	21.5	164	0.48	213	22.7	164.1	0.49
7	228	30.7	189.1	0.66	211	21.9	163.5	0.41	217	23	166.5	0.43
8	230	29.3	179.4	0.62	219	22	161	0.42	228	22.2	164.6	0.43
9	238	28.2	188.1	0.68	234	20.8	163.8	0.32	235	21.5	164	0.34
10	254	29.2	186.7	0.63	241	20	159	0.46	243	21.2	155.6	0.49

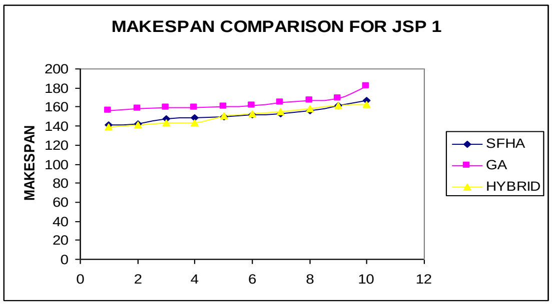

Figure 2 Makespan Comprison for JSP 1

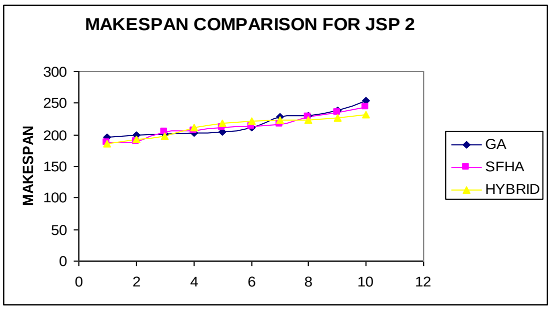

Figure 6 Makespan Comprison for JSP 2

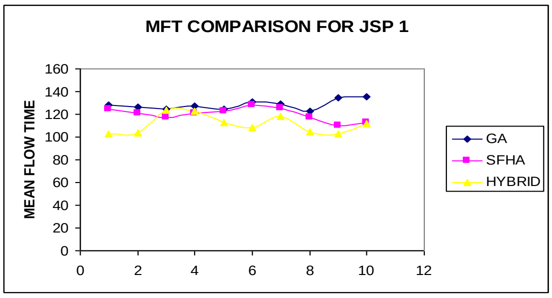

Figure 3 Mean Flow Time Comprison for JSP 1

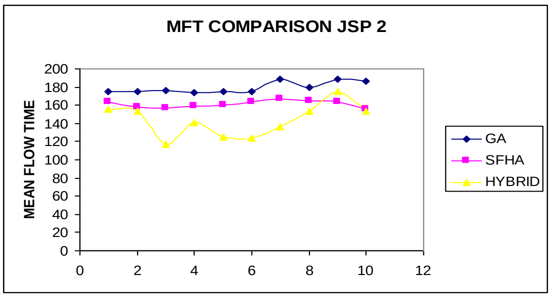

Figure 7 Mean Flow Time Comprison for JSP 2

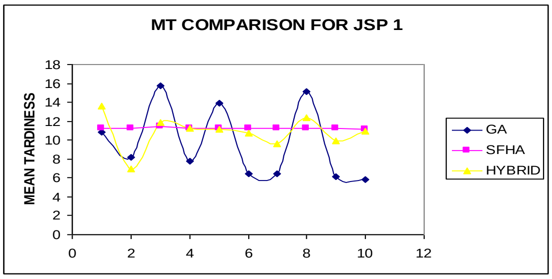

Figure 4 Mean Tardiness Comprison for JSP 1

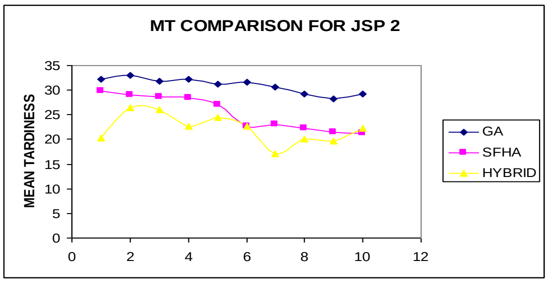

Figure 8 Mean Tardiness Comprison for JSP 2

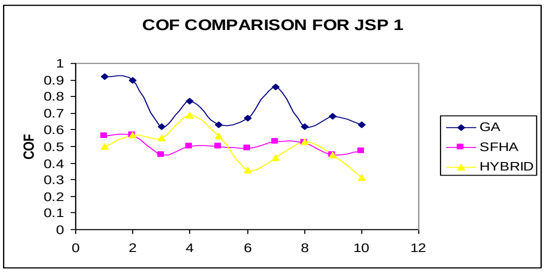

Figure 5 COF Comprison for JSP 1

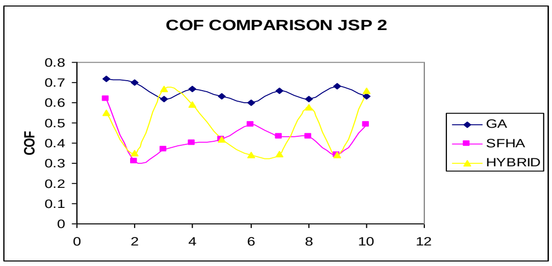

Figure 9 COF Comprison for JSP 2