International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 7 (2015) pp. 18825-18835
© Research India Publications
http://www.ripublication.com

Tricriteria Permutation Flow Shop Scheduling Problem Using Metaheuristic

M. Saravanan, S. Joseph Dominic Vijaya kumar*, R. Srinivasan and S. Paul Singarayar

[1]Principal, Sri Subramanya College of Engineering & Technology, Palani, Tamilnadu, India.

**Assistant Professor, RVS College of Engineering & Technology, Dinidigul, Tamilnadu, India*

[3] Associate Professor, RVS College of Engineering & Technology, Dinidigul, Tamilnadu, India

[3]Assistant Professor, NPR College of Engineering & Technology, Natham, Tamilnadu, India

*[1]drmsaravanan@yahoo.com, *[2]joedominic_s@yahoo.co.in,*
[3]sriparam_2000@yahoo.com, paulsam74@ymail.com

Abstract

The Permutation Flow shop Scheduling Problem is a typical combinational optimization problem and has been proved to be strongly NP- hard. This paper presents an Artificial immune system algorithm meta-heuristic for solving the permutation flow shop scheduling problem to determine the optimal schedule, minimizing the three objectives such as weighted total tardiness, total earliness and makespan. The proposed approach is in conjunction with the constructive heuristic of Nawaz et al. evaluated using benchmark problems taken from Taillard. A computational analysis up to 200 jobs and 20 machines problems has been conducted to evaluate the performance of the approach. Computational experiments indicate that the proposed AIS algorithm is a feasible and effective approach for the multiobjective problem.

Keywords: Earliness, Makespan, Tardinss Artificial immune system, B-Grasp, Simulated annealing.)

Introduction

In a standard flow shop scheduling, there is one machine at a stage and they are arranged perpetually. The jobs are processed on stages sequentially. A Permutation Flow Shop (PFSP) is a modification of standard flow shop, in which n jobs are to be processed on m machines in which each job has one operation at each machine and all

jobs have same processing sequence at every machine. In today's competitive market, the industries have to satisfy their customers with good quality as well as on time delivery. Hence, scheduling must be effective to meet the factory utilization to improve the productivity and also the due date to satisfy the customer. Since delayed due dates can affect the customer relation and the earlier due dates may increase the inventory cost of the finished product. So it becomes necessary that the jobs should be completed on their assigned due dates as possible [1]. Most of the earlier researchers concentrated bi-criteria problems based on flow time and makespan to improve the productivity and factory utilization problem [2,3,4,5]. The makespan criterion is a firm-oriented performance measure. Most of the scheduling research has focused bi-criteria problems and the criteria are related to productivity and facility utilization. Recent researchers include the due date as another criteria and the problem becomes tri-criteria problem [6,7,8,9 10]. E/T is considered as an important measure for due date problems hence in the present work a tri-criteria problem including makes pan, total tardiness and earliness with setup time is solved for effective scheduling.

The multi-criteria scheduling problems are classified into three categories such as i) considering one criterion as objective and the other as constraint. ii) both the criteria as objectives with equal weightage. iii) the criteria are weighted constantly. The problem recognized in this paper has a place with the third class. Setup incorporates work to set up the machine, process, or bench for product parts or the cycle. This incorporates getting tools, positioning work-in-process material, return tooling, cleaning up, setting the required jigs and fixtures, adjusting tools, and inspecting material. Setup times include non-productive operations that must be performed on machines and that are not some part of the job's processing times. These may incorporate, however are not restricted to, cleaning, fixing and releasing parts to machines. In spite of the fact that on a few events setup times could be incorporated in the processing times, in the majority of modern settings it is not conceivable to ignore them. Scheduling problems including setup times might be classified into two classes, the first class is sequence independent and the second is sequence-dependent setup times. The time duration between current and the immediate preceding job is considered for sequence-dependent and for sequence-independent the time depends only on the current job [11, 12, 13,14,15]. Meta-heuristic algorithms have been employed to solve PFSP since it is a typical combination optimization problems and strongly NP- hard [15]. Among various meta-heuristics AIS is a newly evolving algorithm and has been operated successfully to various optimization problems such as [16, 17, 18, 19]. In this work, an AIS algorithm is proposed and evaluated with the problems taken from Taillard and the best of our knowledge this has not been attempted earlier.

The majority of the literature concentrates on a single criteria scheduling problem. Since the criteria are conflicting to each other, it is necessary to obtain trade-off solutions among these objectives and simultaneously rather than going for single objective individually. But in an industrial environment, single criteria will not be much realistic compared with multi-objective.

In this paper, a multi-criteria scheduling problem with separate setup times on permutation flow shop is taken for consideration. The objective function of the

problem is to minimize weighted tardiness, earliness and makespan with setup time. To the best of our knowledge this is the first study that addresses AIS algorithm to minimize the above said objective function.

Problem Statement

The objective of this work is to determine the optimal schedule that minimizes three performance measures such as minimizing weighted sum of total tardiness, total earliness and makespan. The makespan (C_{max}), defined as max (C_1……. C_n) is equivalent to the completion time of the last job to leave the system. Total tardiness (T_j) is a due date related performance measure and it is considered as summation of tardiness of individual jobs, where $T_j = C_j - d_j$. Total earliness (E_j) is also a due date related performance measure but reflects early delivery of jobs and it is considered as summation of earliness of individual jobs, where, $E_j = C_j - d_j$. Therefore, the formulation of the multi-criteria objective function is framed as

$$\text{Min} \quad \lambda_1 \sum_{j=1}^{n} T_j + \lambda_2 \sum_{j=1}^{n} E_j + \lambda_3 C_{\max},$$

where $\lambda 1 + \lambda 2 + \lambda 3 = 1$. Eq.1

Let

i Machine
j Job
m Number of machines
n Number of jobs
Cj Completion time of job j
dj Due date of job j
$\lambda 1$ Weight for Total Tardiness
$\lambda 2$ Weight for Total Earliness
$\lambda 3$ Weight for Total Tardiness

Artificial Immune System

AIS is a evolutionary algorithm inspired by the principle of biological immune system. It has two basic principles Clonal selection and affinity maturation [20]. These two principles are introduced in the algorithm for finding solutions to a wide class of complex problems. The Clonal selection, affinity of the antibody is referred as objective value. Since the affinity value decides the cloning of antibodies, lower the objective value more clones will be generated. The affinity maturation process includes two off- spring mechanisms ie, hyper mutation and receptor editing used for selecting the best from the population.

Artificial immune systems use different immunological mechanisms to solve computational problems [21]. Many immune algorithms have been developed with the aim of finding solutions to a wide class of complex problems. AIS applications include the following areas: clustering and classification [22], anomaly detection [23],

optimization [24], control [25], computer security, learning, bioinformatics, image processing, robotics, virus scanning and web mining [26] and scheduling [27].

Artificial Immune System STEPS

The proposed AIS steps are as follows:

Initialization:

Population size (P), No. of iterations (N) and Amount of low affinity antibodies to be replaced (R).

Objective function evaluation: Calculate the fitness function for each antibody.

Affinity evaluation: calculate the affinity value of each antibody as:

Affinity = (1/objective value) Eq.2

Cloning process: generate copies of the antibodies.

Mutation process (for each clone)

Inverse mutation (generated new antibody).

Decode the new antibody.

Calculate the objective value of the new antibody.

If objective value (new antibody) < objective value (clone) then clone = new antibody else, do pair wise interchange mutation (generate a new antibody)

Decode the new antibody.

Calculate the objective value of the new antibody.

If objective value (new antibody) < objective value (clone) then clone = new antibody

else, clone = clone.

Receptor editing: Then R% of the solutions which has the highest objective value in the population is replaced in the R% of randomly generated solutions.

Termination test: check the stopping criterion. If it is met, return the best antibody; else go to step 2.

Simulated Annealing Algorithm

Simulated Annealing (SA) is a search process that has its origin in the fields of materials science and physics and it was developed for combinatorial optimization problems by Kirkpatrick et al.[28].Simulated Annealing is a combinatorial optimization techniques based on random evaluations of the objective function in such a way that transitions out of a local minimum are possible. The algorithm begins with an initial point and a high temperature ‘T’. A second point is created at random in the vicinity of the initial point and the difference in the function value (ΔE) at these two points is accepted; otherwise the point is accepted with a probability ($-\Delta E/T$). This completes the one iteration of the SA procedure, in the next generation, another point is created at random in the neighborhood of the current point and the metropolis algorithm is used to accept or reject the point. In order to simulate the thermal equilibrium at every temperature, a number of points (n) are usually tested at a particular temperature, before reducing the temperature. The algorithm is terminated

when a sufficiently small temperature is obtained or a small enough function values obtained.

Grasp Approach

GRASP is an iterative constructive meta-heuristic process developed by Feo T [29] in the year 1995. The algorithm comprises two phases: a construction and local search. The construction phase is used for obtaining local optima. In this phase, a greedy randomized algorithm is employed for generating a feasible solutions by evaluating the neighborhood until local minimum is reached. In local search phase the final solution is obtained from these local minimum results.

Procedure GRASP (Max_Iterations, Seed)

1. Read_ Input( );
2. For k=1,…,Max_ Iterations do.
3. Solution ← Greedy _Randomized_ Construction (Seed)
4. Solution ← Local _Search (Solution)
5. Update_ Solution (Solution, Best _ Solution);
6. end;
7. return Best _ Solution; and end GRASP.

Fig 1. Pseudo – code for GRASP

General Structure of GRASP

GRASP–Construction Phase

GRASP construction phase has been developed by Feo and Resende [15] . In every iteration of this phase, greedy function is employed to order the element in a candidate list. A greedy parameter (α) is ranged between zero and one is obtained experimentally. The selection of an element may bring some changes and benefits and the same is reflected in the updation of the next element. This involves the heuristic an adaptive one. The grasp is probabilistic as the top candidate is not chosen from the best candidate list instead it is chosen randomly from the list.

Procedure GRASP_ Construction (seed)

1. Solution ← φ;
2. Evaluate the incremental costs of the candidate elements;
3. While solution is not a completion solution do
4. Generate Restricted Candidate List (RCL)
5. Select ‘S’ element randomly from RCL;
6. Solution ← Solution U(S);
7. Reevaluated the incremental costs;
8. end;
9. return solution;

end grasp _ construction

Fig 2. Pseudo – code of GRASP Construction phase.

GRASP–Local Search Phase

The solutions obtained from construction phase are not so much optimal, even concerning basic neighborhoods. The local search phase improves the constructed solution. A local search heuristic algorithm is employed by successively replacing the current solution by a better solution in the neighborhood of the current solution. This process completes when no better solution is found in the neighborhood.

Computational Results

In this section, the results of the computational tests associated with the AIS algorithm are presented. The coding of the proposed algorithm is programmed in JAVA and implemented on a personal computer with Dual Core and 2GB RAM. To test the performance of the AIS, benchmark problems proposed by Taillard (1993) [44] [17] are selected. Three hundred and sixty problem instances with the number of machines ranging from 5 to 20, number of jobs ranging from 20 to 200 with three sets of weights (0.25,0.25,0.5), (0.5,0.25, 0.25) and (0.33,0.33,0.33) have been generated. Nine different sized benchmark problems each with 10 different instances were considered.

Table 1. lists the nine different benchmark problems and the average value of objective value for 10 different instances for three sets of weights (0.25,0.25,0.5), (0.5,0.25, 0.25) and (0.33,0.33,0.33) and the same plotted in the Fig 1. The results show that results (0.25, 0.25, 0.5) gives lower average value compared to the other two weightage values ranges.

Table 1: Relative percentage deviation values for the weightage ranges (0.5, 0.25, 0.25)

N	**m**	**Average Objective value [sec]**		
		AIS	***B-GRASP***	***SAA***
20	5	0.17425	0.35264	3.67412
20	10	0.29256	0.36589	3.12456
20	20	0.41854	0.45782	4.01245
50	5	0.88742	1.12546	2.74512
50	10	0.69852	1.23213	2.01245
50	20	0.74253	1.51023	2.19854
100	5	0.89568	1.52123	3.55214
100	10	0.60213	1.31024	2.85647
100	20	1.23657	2.32541	4.65892
200	5	1.78546	2.01475	3.34512
200	10	2.21546	2.45216	3.21451
200	20	2.54789	2.98564	4.98521

Table 2: Relative percentage deviation values for the weightage ranges (0.25, 0.25, 0.5)

N	**m**	**Average Objective value [sec]**		
		AIS	***B-GRASP***	***SAA***
20	5	0.18524	0.39473	3.98562
20	10	0.27451	0.35503	3.45213
20	20	0.40123	0.46532	4.16589
50	5	0.79742	1.19854	3.89542
50	10	0.68523	1.25412	2.11045
50	20	0.713643	1.49856	2.21235
100	5	0.87586	1.62458	2.99102
100	10	0.59874	1.41263	3.01236
100	20	1.18542	3.01236	4.23154
200	5	1.52314	2.23658	3.55263
200	10	2.0124	2.64213	3.71254
200	20	2.24561	3.3564	4.78563

Table 3: Relative percentage deviation values for the weightage ranges (0.33, 0.33, 0.33)

N	**m**	**Average Objective value [sec]**		
		AIS	***B-GRASP***	***SAA***
20	5	0.19856	0.40122	4.02136
20	10	0.26584	0.38569	3.65241
20	20	0.39856	0.45123	4.08562
50	5	0.80125	1.32547	3.98562
50	10	0.69472	1.37856	2.36549
50	20	0.75869	1.65427	2.21456
100	5	0.85689	1.85467	3.012365
100	10	0.60123	1.54679	3.22547
100	20	1.5246	3.22134	3.99658
200	5	1.96582	2.54612	3.62143
200	10	2.22145	2.87465	3.98546
200	20	2.87456	3.87456	4.95623

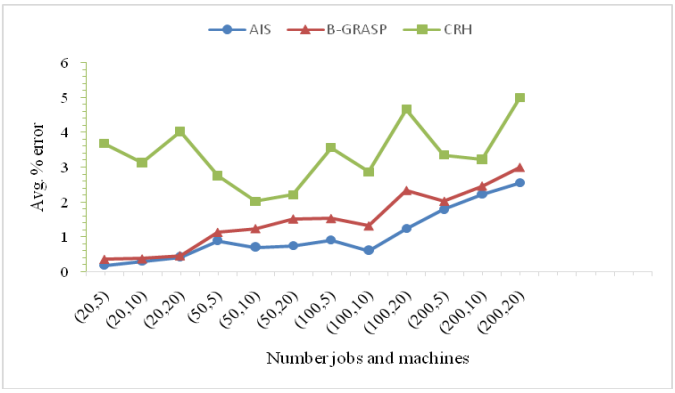

Figure 1: Average objective value vs. number of jobs and machines for Λ= 0.5, 0.25, 0.25

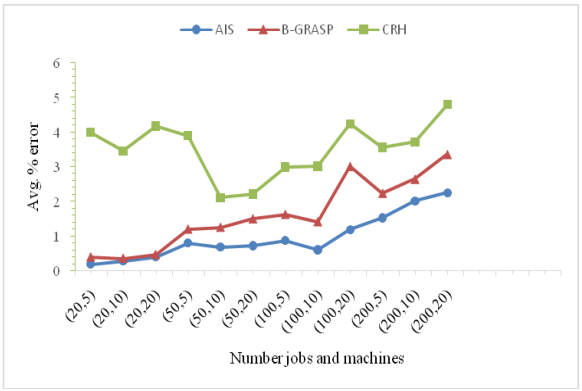

Figure 2: Average objective value vs. number of jobs and machines for Λ= 0.25, 0.25, 0.5

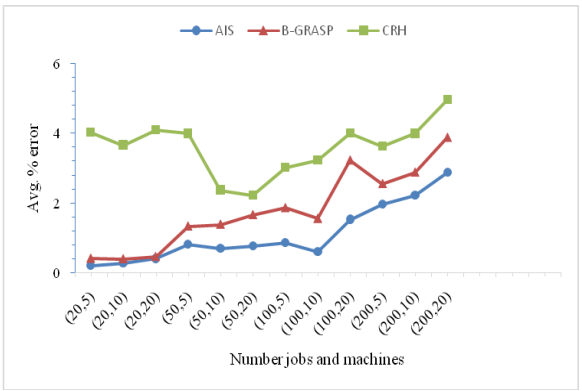

Figure 3: Average objective value vs. number of jobs and machines for Λ= 0.33, 0.33, 0.33

Conclusion

In this study, the permutation flow shop scheduling problem with the tri-objective of minimizing the weighted sum of total tardiness, total earliness and makespan is addressed. The AIS algorithm is tested using Taillard benchmark scheduling problem instances. The performance of this algorithm is compared with B- GRASP and SAA approaches using average value of RPD. The AIS algorithm outperforms with the B-GRASP and simulated annealing algorithm in all the cases. When comparing the performance based on weightage factors in AIS approach, each weightage factor performs better than the other depending on the problem size. For instance, α =.0.5,0.25,0.25, α = 0.25,0.25,0.5, α = 0.33,0.33,0.33 gives lower average RPD value for (n= 100 and 200) problems, (n=50) problems, (n= 20) problems respectively. Moreover α = 0.25,0.25,0.5 has a better performance in most of the cases and in few cases it is equally good with the other two weightage values. Hence it is suggested that AIS algorithm with a weightage factor of α = 0.25 for total tardiness, α = 0.25 for total earliness and α = 0.5 for makespan can provide better results for permutation flow shop scheduling problem with the tri-objective of minimizing tardiness, earliness and makespan. The AIS is proven to be effective and found to be suitable for small-size as well as large-size permutation flow shop scheduling problems.

References

[1] Sung,C.S. and Min,J.I.(2001) 'Theory and methodology scheduling in a two machine flowshop with batch processing machine(s) for earliness measure under a common due date', European Journal of Operational Research, Vol.131, No. , pp. 95-106.

[2] Shih-Wei Li. and Kuo-Ching Ying. (2013) 'Minimizing makespan and total flowtime in permutation flowshops by a bi-objective multi-start simulated-annealing algorithm', Computers & Operations Research, Vol. 40, No. , pp. 1625-1647. Issue 6.

[3] Pasupathy,T., Chandrasekharan Rajendran and Suresh R.K., (2006) 'A multi-objective genetic algorithm for scheduling in flow shops to minimize makespan and total flow time', International Journal of Advanced Manufacturing Technology, Publisher: Springer-Verlag London Ltd, vol. 27, No. , pp. 804-815.

[4] Ravindran,D., Noorul Haq ,A., Selvakumar,S.J. and Sivaraman,R., (2005) 'Flow shop scheduling with multiple objective of minimizing makespan and total flow time', International Journal of Advanced Manufacturing Technology, vol.25, No. ,pp. 1007–1012.

[5] Yagmahan,B., Yenisey, MM. (2010) 'A multi-objective ant colony system algorithm for flow shop scheduling problem', Expert Syst Appl, vol. 37, No. , pp.1361–1368.

[6] Rajendran,C.,(1995) 'Theory and methodology heuristics for scheduling in flow shop with multiple objectives', .European Journal of Operation Research, Vol.82,(3),No. , pp.540 −555.

[7] . Yagmahan,B. and Yenisey, M.M. (2008), 'Ant colony optimization for multi-objective flowshop scheduling problem',Computers and Industrial Engineering, Vol.54,(3), No. , pp. 411-420

[8] Ponnambalam,S.G., Jagannathan,H., Kataria,M. and Gadicherla,A.,(2004) 'A TSP-GA multi- objective algorithm for flow shop scheduling', International Journal of Advanced Manufacturing Technology, Vol.23, No. , pp. 909–915.

[9] Sha D.Y. and Lin H.H.,(2009)'A particle swarm optimization for multiobjective flowshop scheduling', Int J Adv Manuf Technol,Vol. 45, No. , pp.749–758.

[10] Eren,T. and Guner,E.,(2008) 'The triceiteria flowshop scheduling problem', Int J Adv Manuf Technol,Vol.36, No. , pp. 1210–1220.

[11] Yang,W.H. and Liao,C.J.,(1999) 'Survey of scheduling research involving setup times',Int. J. Syst. Sci,Vol. 30,(2), No. , pp. 143–155.

[12] Cheng,T.C.E., Gupta,J.N.D. and Wang,G.,(2000)'A review of flowshop scheduling research with setup times', Product. Operat. Manage,Vol.9,(3), No. , pp. 262–282.

[13] Allahverdi,A., Ng,C.T., Cheng,T.C.E. and Kovalyov,M.Y.,(2008) 'A survey of scheduling problems with setup times or costs', Eur. J. Oper. Res,Vol.187,(3), No. , pp. 985– 1032.

[14] Eren,T. and Guner,E.,(2006) 'A bicriteria flowshop scheduling problem with setup times',Appl. Math. Comp,Vol. 183,(2), No. ,pp.

[15] Rinnooy Kan,A.H.G.(1976) 'Machine scheduling problems: Classification, complexity, and computations. The Hague: Martinus Nijhoff,

[16] Reisi,M. and Moslehi,G.,(2011) 'Minimizing the number of tardy jobs and maximum earliness in the single machine scheduling using an artificial immune system', Int J Adv Manuf Technol,Vol. 54, No. , pp, 749–756.

[17] Engin,O. and Doyen,A.,(2004) 'A new approach to solve hybrid flow-shop scheduling problems by artificial immune system', Futur Gener Comput Syst,Vol. 20, No. , pp.1083–1095.

[18] Zandieh,M., Fatemi Ghomi,S.M.T. and Moattar Husseini,S.M.,(2006) 'An immune algorithm approach to hybrid flow shops scheduling with sequence dependent setup times', Appl Math Comput,Vol. 180,(1), No. , pp. 111–127.

[19] Bagheri,A., Zandieh,M., Mahdavi,I. and Yazdani,M.,(2010)'An artificial immune algorithm for the flexible job-shop scheduling problem ', Futur Gener Comput Syst,Vol. 26, No. , pp. 533–541.

[20] Leandro de Castro,N., Fernando,J., and Von Zuben.(2002), 'Learning and optimization using the Clonal selection principle',IEEE Transactions on Evolutionary Computation,Vol. 6,(3), No. , pp. 239–351.

[21] Dasgupta,D. and Nin˜ o,L.F.,(2009) 'Immunological computation theory and applications. Boca Raton: CRC Press/Taylor & Francis Group.

[22] Watkins,A. and Timmis,J.,(2004) 'Artificial immune recognition system (airs): an immune- inspired supervised learning algorithm', Journal Genetic Programming and Evolvable Machines,Vol.5, No. , pp.291–317.

[23] Dasgupta,D. and Forrest,S.,(1995)'Novelty detection in time series data using ideas from immunology', In: Presented at the proceedings of the 5th international confer- ence on intelligent systems.

[24] De Castro,L.N. and Timmis,J.,(2002) 'An artificial immune network for multimodal func- tion optimization', In: Presented at the proceedings of the 2002 congress on evolutionary computation.

[25] Kim,D.H. and Cho,J.H.,(2004)'Intelligent tuning of PID controller with disturbance function using immune algorithm',In: Presented at the IEEE international conference on computational intelligence for measurement systems and applications.

[26] Hart,E. and Timmis,J.,(2008) 'Application areas of AIS: the past, the present and the future', Applied Soft Computing,Vol.8, No. , pp.191–201.

[27] Li,B., Wu,S., Yang,J., Zhou,Y. and Du,M.,(2012) 'A three-fold approach for job shop problems: a divide-and-integrate strategy with immune algorithm', Journal of Manufacturing Systems,Vol.3, No. , pp.1195–203.

[28] Kirkpatrick,S., Gelatt,C. and Vecchi,P., (1983) 'Optimization by simulated annealing', Science,Vol. 220, No. , pp.671–679.

[29] Feo,T. and Resende,M.G.C., (1995) 'Greedy randomizes adaptive search procedures', J Glob Optim,Vol. 6, No. , pp.109–133

[30] Taillard,E.,(1993) 'Benchmark for basic scheduling problems',European Journal of Operational Research,Vol. 64, No. , pp. 278-285.