VOL. 10, NO. 13, JULY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

GT HEURISTIC FOR SOLVING MULTI OBJECTIVE JOB SHOP SCHEDULING PROBLEMS

M. Chandrasekaran[1], D. Lakshmipathy[1] and P. Sriramya[2]
[1]Department of Mechanical Engineering, Vels University, Chennai, India
[2]Department of Computer Science Engineering, Saveetha School of Engineering, Chennai, India
E-Mail: ch_sekaran@yahoo.com

ABSTRACT

The n-job, m-machine Job shop scheduling (JSP) problem is one of the general production scheduling problems in manufacturing system. Scheduling problems vary widely according to specific production tasks but most are NP-hard problems. Scheduling problems are usually solved using heuristics to get optimal or near optimal solutions because problems found in practical applications cannot be solved to optimality using reasonable resources in many cases. In this paper, optimization of three objectives mean job flow time, mean job tardiness and makespan are considered. New Game theory based heuristic approach (GT) is used for finding optimal makespan, mean flow time, mean tardiness values of different size problems and generating a feasible schedule that minimizes three objectives. This schedule is called a Pareto optimal solution. A set of non-dominant solutions have been identified using Naive and Slow approach. The findings are compared with Genetic Algorithm (GA) that tested the same problems. The proposed GT heuristic is competent and proves to be a good problem-solving technique for multi objective optimization in job shop scheduling.

Keywords: job shop scheduling, multi objectives, GT heuristic, pareto optimal solution, naïve, slow approach.

1. 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 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. LITERATURE REVIEW

a) Job shop scheduling

Bruker [1] show that the Job Shop scheduling Problem (JSP) is an NP-hard combinatorial problem. Because of the NP-hard characteristics of job shop scheduling [2], 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]. 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 time based optimization in Daniels [8], Lee and Jung [9], and Murata, Ishibuchi, and Tanaka [10].

b) Graph Theory based (GT) heuristic

Trees are more important data structures which come in many forms. Sometimes trees are static in the sense that their shape is determined before running of the algorithm, and they do not change shape while the algorithm runs. In other cases, trees are dynamic, meaning that they undergo shape changes during the running of the algorithm. In GT heuristic root node means origin or stating node. Children mean sub nodes in the parent. Leaf node means node with our children. Parent node means node with children.

The remainder of the paper is organized as follows. Section 3 describes the Problem formulation. Section 4 introduces the new GT heuristic for JSP problem. Section 5 shows Implementation of GT heuristic. Section 6 shows the Results and discussion. Finally, section 7 presents the conclusion of this work.

3. PROBLEM FORMULATION

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,

VOL. 10, NO. 13, JULY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

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. PROPOSED NEW HEURISTIC FOR MULTI OBJECTIVE JSP PROBLEM

GT Heuristic for solving JSP

Step-1: Initialization

Initial sequences are generated randomly. Objective values for initial sequences are calculated. Roulette wheel method is used for ranking the sequences.

Step-2: Depth First Search

In depth first search, initial sequences are changed in tree structure format. Based on the tree, the sequence generated by traversing root node then leaf node and finally parent node. In this order, sequence is traversed and the corresponding objective values are calculated.

Step-3: Breadth First Search

Based on the tree, the sequence generated by traversing root node then parent node and finally leaf node. In this order, sequence is traversed and the corresponding objective values are calculated.

Step-4: Tree traversal

In computer science, tree traversal is the process of visiting each node in a tree data structure. Tree traversal, also called walking the tree, provides for sequential processing of each node in what is, by nature, a non-sequential data structure.
Such traversals are classified by the order in which the nodes are visited.

i) Pre-order traversal: Each node is visited before any of its children.

ii) Post-order traversal: Each node would be visited after all of its children. In both cases, values in the left subtree are printed before values in the right subtree.

iii) In-order traversal: Visits each node between the nodes in its left subtree and the nodes in its right subtree. This is a particularly common way of traversing a binary search tree, because it gives the values in increasing order.

If n is a node in a binary search tree, then everything in n's left subtree is less than n, and everything in n's right subtree is greater than or equal to n. Thus, the left subtrees in order, using a recursive call, and then visit n, and then visit the right subtree in order. Assume the recursive calls correctly visit the subtrees in order using the mathematical principle of structural induction. Flow chart of GT Heuristic is shown in Figure-1.

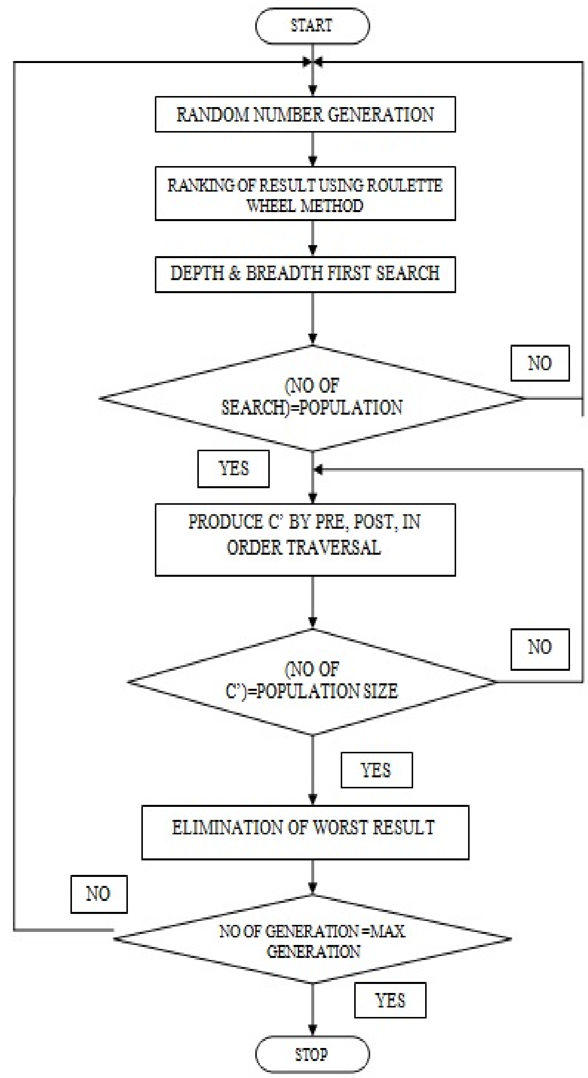

Figure-1. Flow chart of GT heuristic.

5. IMPLEMENTATION OF GT HEURISTIC (GT)

a) GT heuristic parameters

The following parameters are considered in trail and error basis.
Population size (n) = 10
Fitness function = makespan
Root node = origin or starting point node
Leaf node = node without children
Parent node = node with children on both side
Ranking method = Roulette Wheel Method
Non-dominated set = Naive and Slow approach

VOL. 10, NO. 13, JULY 2015 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

Termination Criteria = Number of iterations (100)

b) Numerical illustration

Test Problems JSP1 & JSP2 is taken from Bagchi [7] for evaluating the performance of GT Heuristic

Step 1: Initialization

Initial sequences for JSP1 problem is generated randomly. Table-1 shows the initial sequence with its objective values. Roulette wheel method is used for ranking the sequences.

Table-1. Initial sequence.

10	7	9	5	1	10	4	6	2	8	Makespan =236 Mean flow time =176.30 Mean tardiness =33.38 COF = 0.6802
3	1	5	3	2	4	1	4	6	5	
7	1	2	7	8	1	7	6	9	8	
6	10	7	5	4	6	3	3	10	9	
2	8	9	5	9	10	8	4	3	2	

Step 2: Depth First Search

In depth first search, initial sequences are changed in tree structure format with 10 as root node. In this search, root node is first selected then nodes without children (leaf nodes) are considered and finally nodes with children (parent nodes) are taken for traversing process. Sequence is traversed and the corresponding objective values are calculated. Table-2 gives the values obtained from depth first search process.

Table-2. Depth First Search process

10	7	5	3	2	4	1	6	10	7	Makespan=197 Mean flow time =152.39 Mean tardiness = 31.77 COF = 0.676182
9	8	4	3	2	6	5	9	8	4	
2	1	10	3	5	6	5	7	1	10	
9	8	3	2	4	7	6	8	9	1	
10	6	2	4	3	5	7	9	8	1	

Step 3: Breadth First Search

In breadth first search, sequence is generated with the root node i.e. 10. Parent nodes are considered next and then finally leaf nodes. At the end of Breath First search method, the sequence obtained with objective values are shown in Table-3.

Table-3. Breadth First Search process.

10	1	5	9	7	3	1	8	4	6	Makespan=165 Mean flow time =109.697 Mean tardiness = 26.11 COF = 0.3218
1	2	4	6	4	2	5	2	7	10	
3	6	1	3	6	4	9	7	2	5	
10	5	3	5	8	9	7	7	8	1	
6	8	10	3	9	9	2	4	8	10	

Step 4: In order traversal

Tree consists of Left sub tree and Right sub tree. The traversal is done for by considering left sub tree nodes then the root node and finally the right sub tree nodes. In order traversal process results are reported in Table-4.

Table-4. In order traversal process.

1	8	9	7	5	3	4	2	6	10	Makespan =187 Mean flow time =138.00 Mean tardiness = 24.84 COF= 0.5857
1	9	8	6	7	4	2	3	8	9	
10	1	7	5	6	5	3	10	1	2	
4	8	9	5	6	2	3	4	8	9	
7	10	6	1	4	2	3	5	7	10	

Step 5: Post order traversal

The traversal is carried out for traversing the sequence by considering Left Sub tree nodes first then Right Sub Tree nodes and finally Root node. For the traversed sequence the corresponding objective values are calculated. Table-5 illustrates the results obtained from post traversal process.

Table-5. Post traversal process.

2	4	3	8	9	7	10	6	2	3	Makespan = 200 Mean flow time =141.39 Mean tardiness = 25.38 COF = 0.3812
5	6	4	1	5	8	10	1	3	2	
8	9	10	2	4	3	1	7	6	9	
8	7	5	6	5	4	9	7	3	5	
4	2	8	9	7	6	1	10	1	10	

Step 6: Preorder traversal

The traversal is carried out from Root node then Left Sub Tree nodes and Right Sub Tree nodes. After traversing the sequence based on preorder traversal process, the value of makespan, mean flow time and mean tardiness is calculated. Final results obtained after the first iteration is shown in Table-6.

Table-6. Pre traversal process.

10	7	5	3	2	4	1	6	10	7	Makespan=179 Mean flow time = 122.39 Mean tardiness = 13.77 COF =0.3616
9	8	4	3	2	6	5	9	8	4	
2	1	10	3	5	6	5	7	1	10	
9	8	3	2	4	7	6	8	9	1	
10	6	2	4	3	5	7	9	8	1	

Step 7: The final results obtained for JSP1 from GT heuristic is as follows

After 100 iterations, the results obtained by using GT heuristic are as follows:

Table-7. Results from GT Heuristic for JSP1.

Makespan	Mean tardiness	Mean flow time	COF
152	16.845	117.8	0.5811
162	16.805	121.2	0.6450
163	15.802	128.5	0.5674
164	12.97	125.28	0.5714
168	12.042	124.7	0.5396
170	14.985	131.65	0.6167
172	11.442	133.42	0.5061
175	12.804	129.58	0.6412
177	17.822	119.91	0.6829
178	19.759	139.3	0.6323

6. RESULTS AND DISCUSSION

The new heuristic is developed and 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 new GT heuristic 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 GT heuristic minimizes all objectives simultaneously. GT heuristic is compared with the similar previous work using GA [12] and shown in Table-8 and Table-9.

7. CONCLUSIONS

In this paper, a new heuristic 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. This approach uses simple but effective techniques depth first search, Breadth first search and tree traversal. This approach has been tested on JSP 1 and JSP 2 problem instances given in Bagchi [7]. The findings were compared with Genetic Algorithm [12] that tested the same problems. The New heuristic gives better results than the genetic algorithm. The proposed new heuristic is competent and proves to be a good problem-solving technique for job shop scheduling.

Notations Used

MS = Makespan
MFT = Mean Flow Time
MT = Mean Tardiness
COF = Combined Objective Function.

REFERENCES

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

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

[3] Erschler J.F., Roubellat J.P. and Vernhes. 1976. Finding some essential characteristics of the feasible solutions for a scheduling problem, Operations Research, 24, pp. 774-783.

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

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

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

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

[8] Daniels D.L. 1994. Incorporating performance information into multiobjective scheduling. European Journal Operational Research, Vol. 77, pp. 272–286.

[9] Lee S.M. and Jung H.J. 1989. A multi-objective production planning model in A flexible manufacturing environment. International Journal Production Research, Vol. 27, No. 11, pp. 1981–1992.

[10] Murata T., Hisao I. and Tanaka H. 1996. Multi-objective genetic algorithm and its applications to flow shop scheduling. Computer Industrial Engineering, Vol. 30, No. 6, pp. 957–968.

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

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