Annals of Pure and Applied Mathematics
Vol. 7, No. 1, 2014, 29-34
ISSN: 2279-087X (P), 2279-0888(online)
Published on 9 September 2014
www.researchmathsci.org

Annals of
Pure and Applied
Mathematics

Optimization for Flexible Job Shop Scheduling by Evolutionary Representation

K.Bharathi[1] **and** ***C.Vijaylakshmi***[2]

[1] Department of Mathematics, SCSVMV University
Kanchipuram-631561, Tamil Nadu, India
Email: 03bharathi@gmail.com

[2] SAS, Mathematics Division, VIT University
Chennai- 600 127, Tamil Nadu, India.
Email: vijusesha2002@yahoo.co.in

Received 20 June 2014; accepted 4 August 2014

Abstract. In this paper we have discussed a flexible job-shop scheduling problem by considering the cases as the assignment of each operation to a machine, and the other is the scheduling of this set of operations in order to minimize our criterion (e.g. the make span and completion time of each job). After applying the operators like crossover and mutation criterion where minimized. Here, we propose effective genetic encodings, such as job and machine representations as matrices of the chromosome, and Genetic operators where associated with these representations.

Keywords: Flexible job-shop scheduling, Optimization, Evolutionary Algorithms, Genetic representation, Encoding matrices, Makespan.

AMS Mathematics Subject Classification (2010): 05C78

1. Introduction

Job shop scheduling problems (JSS) are computationally complex problems. Because JSS are NP-hard i.e., they can't be solved within polynomial time force or undirected search methods are not typically feasible, at least for problems of any size. Thus JSS tend to be solved using a combination of search and heuristics to get optimal or near optimal solutions. The term 'Scheduling' in manufacturing systems is used to the determination of the sequence in which parts are to be processed over the production stages, followed by the determination of the start-time and finish-time of processing of parts, so as to meet an objective or a set of objectives.

Scheduling plays a crucial role to increase the efficiency and productivity of the manufacturing system. The scheduling can be classified into (i) Single machine scheduling (ii) Flow shop scheduling (iii) Job shop scheduling. Optimisation methods attempt to find the optimal solution through mathematical programming techniques or methods [5-7]. However, mathematical programming methods are time-consuming, and thus, many researchers focus on developing heuristic algorithms [11-15], algorithms in common use include shifting bottleneck (SB) [15], Tabu search (TS) [10], simulated annealing (SA) [9], the genetic algorithm (GA) [8] , artificial immune system (AIS) [16], and modified Particle swarm optimization (PSO)[17]. In [19] proposed a local search genetic algorithm that uses an efficient solution representation strategy in which both checking of the constraints and repair mechanism can be avoided.

2. Structure of the scheduling problem

Consider a set of ‘n’ jobs {J_i}, $1 \le i \le n$; these jobs are independent of one another. Each job ‘J_i’ has an operating sequence, called P_i. Each operating sequence P_i is an ordered series of X_i operations, O_{ij} indicating the position of the operation in the technological sequence of the job. The realization of each operation O_{ij} requires a resource or a machine selected from a set of machines,{M_k},$1 \le k \le m$; ‘m’ is the total number of machines existing in the shop, this implying the existence of an assignment problem. There is a pre-defined set of processing times; for a given machine, and a given operation, the processing time is denoted by $T_{i,j,M}$. An operation which has started runs to completion (non-preemption condition). Each machine can perform operations one after another (resource constraints). The time required to complete the whole job constitutes the make span C_{max}. The time required to complete each jobs is as CT_i our objective is to determine the set of completion times of all jobs to minimize C_{max} and also to minimize CT_i .

3. Representation of the solution

The chromosome is represented by a set of jobs and each job is a matrix which contains its assignment operations. These operations are represented by three terms. The first column is the order number of the machine in its operating sequence. The second is the starting time of the operation if its assignment on this machine. The third is the completion time of the operation if its assignment on this machine. That is

$$J_K = \begin{bmatrix} j & S_{ij} & C_{ij} \\ j' & S'_{ij} & C'_{ij} \\ \dots & \dots & \dots \end{bmatrix}, \text{ where } k, i = 1, 2, \dots, n \text{ and } j = 1, 2, \dots, m.$$

4. Numerical calculation

Three jobs and five machines are considered. The operating sequences of these jobs are as follows in table 1.

J	O	M1	M2	M3	M4	M5
J_1	$O_{1,1}$	1	8	3	7	5
	$O_{2,1}$	3	5	2	6	4
	$O_{3,1}$	6	7	1	4	3
J_2	$O_{1,2}$	1	4	5	3	8
	$O_{2,2}$	2	8	4	9	3
	$O_{3,2}$	9	5	1	2	4
J_3	$O_{1,3}$	1	8	9	3	2
	$O_{2,3}$	5	9	2	5	3

Table 1: The operating sequences

According to the machine used, the processing time of operations is described. One of the solution to the problem is as the matrix representation in fig. 1. Where C_{max} = 11

$$\begin{array}{ccc} J_1 & J_2 & J_3 \\ \begin{bmatrix} 3 & 0 & 3 \\ 3 & 3 & 5 \\ 4 & 5 & 9 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 5 \\ 3 & 5 & 6 \end{bmatrix} & \begin{bmatrix} 5 & 0 & 2 \\ 2 & 2 & 11 \\ 0 & 11 & 11 \end{bmatrix} \end{array}$$

Figure 1: Example

5. Evolutionary representation

5.1. Initial population

The initial population is usually chosen at random. But in a combinatorial problem such as job shop scheduling, some constraints such as precedence and resources constraints must be satisfied. In this case, the binary representation is not convenient and chromosome syntax must be found to fit the problem. For these reasons, we have designed a matrix representation of the chromosome, and in order to create and to permit our set of solutions to evolve in a very large domain, we shall use a combination of some methods. We use a combination of the following priority rules. SPT: a high priority for the operation that has the Shortest Processing Time. LPT: a high priority for the operation that has the Longest Processing Time. LM: a high priority for the operation that permits to balance the load of the machine. Accordingly two solutions of the example (table 1) are taken as parent fig. 2 with C_{max} = 11 and fig.3 with C_{max} = 13.

$$\begin{array}{ccc} J_1 & J_2 & J_3 \\ \begin{bmatrix} 3 & 0 & 3 \\ 3 & 3 & 5 \\ 4 & 5 & 9 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 5 \\ 3 & 5 & 6 \end{bmatrix} & \begin{bmatrix} 5 & 0 & 2 \\ 2 & 2 & 11 \\ 0 & 11 & 11 \end{bmatrix} \end{array}$$

Figur 2: Parent 1

$$\begin{array}{ccc} J_1 & J_2 & J_3 \\ \begin{bmatrix} 1 & 0 & 3 \\ 4 & 3 & 6 \\ 4 & 6 & 13 \end{bmatrix} & \begin{bmatrix} 4 & 0 & 3 \\ 5 & 3 & 6 \\ 2 & 6 & 11 \end{bmatrix} & \begin{bmatrix} 5 & 0 & 2 \\ 3 & 2 & 4 \\ 0 & 4 & 4 \end{bmatrix} \end{array}$$

Figure 3: Parent 2

5.2. Crossover operator

Crossover involves combining elements from two parent chromosomes into one or more child chromosomes The role of the crossover is to generate a better solution by exchanging information contained in the current good ones. Here we use the folling steps to get the offsprings

STEP 1: For child 1 job 1 of parent 1 is placed and jobs 2and 3 of parent 2 is placed.
STEP 2: For child 2 job 1 of parent 2 is placed and jobs 2and 3 of parent 1 is placed.

The offspring's after crossover where represented as shown in the fig.3 and fig.4 where child 1 having C_{max}=11 and child 2 having C_{max}=13.

$$\begin{array}{ccc} J_1 & J_2 & J_3 \\ \begin{bmatrix} 3 & 0 & 3 \\ 4 & 3 & 5 \\ 4 & 5 & 8 \end{bmatrix} & \begin{bmatrix} 4 & 0 & 3 \\ 5 & 3 & 6 \\ 2 & 6 & 11 \end{bmatrix} & \begin{bmatrix} 5 & 0 & 6 \\ 3 & 6 & 7 \\ 0 & 7 & 7 \end{bmatrix} \end{array}$$

Figure 3: Child 1

$$\begin{array}{ccc} J_1 & J_2 & J_3 \\ \begin{bmatrix} 1 & 0 & 1 \\ 4 & 1 & 7 \\ 4 & 7 & 13 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 4 \\ 3 & 4 & 5 \end{bmatrix} & \begin{bmatrix} 5 & 0 & 2 \\ 2 & 4 & 13 \\ 0 & 13 & 13 \end{bmatrix} \end{array}$$

Figure 4: Child 2

5.3. Mutation

After crossover, each child produced by the crossover undergoes mutation with a low probability. Hear the mutation is down as follows

1. Choose a job with largest completion time CT_i.
2. In that job J_i Replace the machine with largest operating time by least operating time.

The offspring's after mutation where represented as shown in the fig.10 and fig.11 where mutation child 1 having C_{max}=8 and mutation child 2 having C_{max}=5.

$$\begin{array}{ccc} J_1 & J_2 & J_3 \\ \begin{bmatrix} 3 & 0 & 3 \\ 4 & 3 & 5 \\ 4 & 5 & 8 \end{bmatrix} & \begin{bmatrix} 4 & 0 & 3 \\ 5 & 3 & 6 \\ 3 & 7 & 8 \end{bmatrix} & \begin{bmatrix} 5 & 0 & 6 \\ 3 & 6 & 7 \\ 0 & 7 & 7 \end{bmatrix} \end{array}$$

Figure 5: Mutation Child 1

$$\begin{array}{ccc} J_1 & J_2 & J_3 \\ \begin{bmatrix} 1 & 0 & 1 \\ 3 & 1 & 3 \\ 3 & 3 & 4 \end{bmatrix} & \begin{bmatrix} 1 & 1 & 2 \\ 1 & 2 & 4 \\ 3 & 4 & 5 \end{bmatrix} & \begin{bmatrix} 2 & 0 & 2 \\ 2 & 2 & 5 \\ 0 & 5 & 5 \end{bmatrix} \end{array}$$

Figure 6: Mutation Child 2

After the variation operators like crossover and mutation we have got the solution with the makespan as 8 and 5 where the completion time for each jobs CT_i is also minimized.

6. Conclusion

Scheduling can be defined as a problem of finding an optimal sequence to execute a finite set of operations satisfying most of the constraints. The problem so formulated is extremely difficult to solve, as it comprises several concurrent goals and several resources which must be allocated to lead our goals, which are to maximize the utilization of individuals and/or machines and to minimize the time required to complete the entire process being scheduled. Here, we propose effective genetic encodings, such as job and machine representations as matrices of the chromosome, and Genetic operators where associated with these representations.

REFERENCES

1. S.G.Ponnambalam, P.Aravindan and S.V.Rajesh, A tabu search algorithm for job shop scheduling, *International Journal of Advanced Manufacturing Technology*, 16 (2000) 765-771.
2. P.V.Laarhoven, E.Aarts and J.K.Lenstra, Job shop scheduling by simulated annealing. *Operations Research*, 40 (1992) 113-125.
3. M.Kolonko, Some new results on simulated annealing applied to job shop scheduling problem, *European Journal of Operational Research*, 113 (1999) 123-136.

4. R.T.Mogaddam, F.Jolai, F.Vaziri, P.K.Ahmed and A.Azaron, A hybrid method for solving stochastic job shop scheduling problems, A*pplied Mathematics and Computation*, 170 (2005) 185-206.
5. L.Wang and D.Z.Zheng, An effective hybrid optimization strategy for job-shop scheduling problems, *Computers & Operations Research*, 28 (2001) 585-596.
6. J.Zhang, X.Hu, X.Tan, J.H.Zhong and Q.Huang, Implementation of an Ant Colony Optimization technique for job shop scheduling problem, *Transactions of the Institute of Measurement and Control*, 28 (2006) 93-108.
7. K.L.Huang and C.J.Liao, Ant colony optimization combined with taboo search for the job shop scheduling problem, *Computers & Operations Research*, 35 (2008) 1030-1046.
8. J.Montgomery, C.Fayad and S.Petrovic, Solution representation for job shop scheduling problems in ant colony optimization, *Faculty of Information & Communication Technologies,* Swinburne University of Technology, 2006.
9. M.Ventresca and B.Ombuki, Ant Colony Optimization for Job Shop Scheduling Problem, *in Proceedings of 8th IASTED International Conference On Artificial Intelligence and Soft Computing,* 2004, pp. 451-152.
10. S.Petrovic and C.Fayad, A genetic algorithm for job shop scheduling with load balancing, *School of Computer Science and Information Technology,* University of Nottingham, 2005.
11. S.Rajakumar, V.P.Arunachalam and V.Selladurai, Workflow balancing in parallel machine scheduling with precedence constraints using genetic algorithm, *Journal of Manufacturing Technology Management*, 17 (2006) 239-254.
12. B.M.Ombuki and M.Ventresca, Local search genetic algorithms for the job shop scheduling problem, *Applied Intelligence,* 21 (2004) 99-109.
13. Y.Chen, Z.Z.Li and Z.W.Wang, Multi-agent-based genetic algorithm for JSSP, *in Proceedings of the third international conference on Machine Learning and Cybernetics*, 2004, pp. 267-270.
14. L.Asadzadeh and K.Zamanifar, An agent-based parallel approach for the job shop scheduling problem with genetic algorithms, *Mathematical and Computer Modeling*, 52 (2010) 1957-1965.
15. F.Bellifemine, A.Poggi and G.Rimassa, Developing multi-agent systems with a FIPAcompliant agent framework, *Software: Practice and Experience*, 31 (2001) 103-128.
16. L.Asadzadeh, K.Zamanifar, Design and implementation of a multi-agent system for the job shop scheduling problem, *International Journal of Computer Science and Security*, 5 (2) (2001) 287-297.
17. A.Sadrzadeh, Solving the job shop scheduling problem via a modified PSO, *Technical Journal of Engineering and Applied Sciences*, 2013
18. S.Zhang, A sequence list algorithm for the job shop scheduling problem, *The Open Electrical & Electronic Engineering Journal*, 7(1) (2013) 55-61.
19. V.Ravibabu, A novel metaheuristics to solve mixed shop scheduling problems , *International Journal in Foundations of Computer Science & Technology*,3(2) (2013).

20. A.Khadwilard, S.Chansombat, T.Thepphakorn, P.Thapatsuwan, W.Chainate and P.Pongcharoen, Application of firefly algorithm and its parameter setting for job shop scheduling , *The Journal of Industrial Technology*, 8 (2012).