International Journal of Information Technology and Knowledge Management
January-June 2011, Volume 4, No. 1, pp. 105-108

# PERFORMANCE OF GENETIC ALGORITHMS FOR SOLVING FLEXIBLE JOB-SHOP SCHEDULING PROBLEM

Amit Kumar[1] & Rajnesh Singh[2]

A Job-Shop Scheduling is a process–organized manufacturing facility. Its main characteristics are that a great diversity of jobs is performed. A Job-Shop produces goods (parts) and these parts have one or more alternatives process plans. Each process plan consists of a sequence of a operations and these operations require resources and have certain (predefined) duration on machines. The Job-Shop Scheduling is a problem of planning and organization of a set of tasks to be performed on a set of resources with variable performance. In this paper, two approaches Jobs Sequencing List Oriented Genetic Algorithm and Operations machines Coding Oriented Genetic Algorithm have been implemented and compared for solution of the Job-Shop scheduling problem. Each approach has its own coding, evaluation function, crossovers and mutations applicable in Job-Shop scheduling problem to minimize the makespan, the workload of the most loaded machine and the total workload of the machines. Jobs Sequencing List Oriented Genetic Algorithm has been found to be the best out of two approaches to minimize the objectives.

Keywords: Genetic Algorithm, Flexible Job-Shop Scheduling Problem

## 1. Introduction

The job-shop scheduling problem is the selection of a sequence of operations together with assignment of start/end times and machines for each operation.

The problem is to organize the execution of N jobs on M machines. The set of machines is noted U. Each job j represents a number of $n_j$ non preemptable ordered operations. The execution of each operation i of a job j (noted $O_{i,j}$) requires one resource or machine selected from a set of available machines.

In the job-shop scheduling problem following inputs are given:

- Number of machines;
- Number of jobs;
- Number of operations for each job;
- Process time of each operation for each machine.

Objective is to compare two genetic algorithms with different chromosome representations, different evaluation function, different crossovers and mutations and find out which algorithm minimizes better depending upon the following criteria:

1. The makespan, which is the time in which all operations are executed.
2. The workload of the most loaded machine.
3. The total workload of machines.

Constraints:

1. For each job, the order of operations is fixed.
2. 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 is completed.

Hypothesis:

1. All machines are available at t = 0.
2. All jobs can be started at t = 0.

## 2. Genetic Algorithm

The Genetic Algorithm is a stochastic global search method that imitates this process of natural biological evolution, operating on “populations” of potential solutions by applying the law of the jungle where the survival is for the fittest, hopefully producing better approximations to a given application’s solution. Until a stopping criterion is reached (e.g. certain number of generations or a mean deviation in the population), a new set of approximations is created at each generation.

## 3. Result

The two algorithms have been implemented and tested to compare the performance. 15 different runs were performed

[1] Department of Information Technology, IEC College of Engineering and Technology, Greater Noida, India

[2] Department of Information Technology, IIMT College of Engineering, Greater Noida, India

Email: [1]amitvnskumar@rediffmail.com, [2]rajneshcdc.mtech@gmail.com

in [9] JSL oriented GA and [5] OMC oriented GA with different initial populations. To be fair each run of [9] JSL oriented GA and [5] OMC oriented GA started with the same initial population. The reproduction was permitted up to 800 generations. Crossover-rate is 0.9 and mutation-rate 0.02. Population size is 40. The [9] JSL oriented GA terminated when makespan, the workload of most loaded machine and the workload of machine at generation g and generation g-100 are same. [5] OMC oriented GA terminated when makespan at generation g and generation g-100 are same.

The table 5.1 shows the average generation number, average makespan, average workload of most loaded machine and average total workload of the machines over 15 runs because the performance of each approach at each run is not same.

For workload of the most loaded machine, JSL oriented GA outperformed the OMC oriented GA.

OMC oriented GA is good when makespan is only considered. When all three objectives makespan, workload of the most loaded machine and total workload of the machine are considered, JSL oriented GA outperformed OMC oriented GA in terms of minimizing the objectives but converged late.

| | Input First Processing Time | | | |
|---|---|---|---|---|
| | m1 | m2 | m3 | m4 |
| Job_1 | | | | |
| Operation_1 | 3 | 1 | 1 | 2 |
| Operation_2 | 3 | 5 | 1 | 2 |
| Job_2 | | | | |
| Operation_1 | 4 | 1 | 1 | 4 |
| Operation_2 | 1 | 2 | 2 | 4 |
| Job_3 | | | | |
| Operation_1 | 3 | 5 | 7 | 6 |
| Operation_2 | 4 | 5 | 6 | 7 |
| Operation_3 | 4 | 2 | 8 | 3 |
| Job_4 | | | | |
| Operation_1 | 7 | 3 | 9 | 3 |
| Operation_2 | 1 | 2 | 4 | 3 |
| Operation_3 | 2 | 1 | 6 | 7 |

| Approaches | JSL oriented GA | | | OMC oriented GA | | |
|---|---|---|---|---|---|---|
| Average Result & Generation number | Makespan | Work Load of the most loaded machine | Total work load of the machines | Makespan | Work Load of the most loaded machine | Total work load of the machines |
| Average Result to first input | 9.8 | 7 | 21 | 9.2 | 8.6 | 22.8 |
| Average Generation Number to the first input | | 48.4 | | | 32.06 | |
| Average Result to second input | 4.8 | 3 | 28 | 4.8 | 4.4 | 30 |
| Average Generation Number to second input | | 138.8 | | | 111.2 | |
| Average Result to third input | 5 | 3.8 | 42.6 | 5 | 4.4 | 46.2 |
| Average Generation Number to third input | | 294.8 | | | 134.8 | |

Fig. 3.1: Average Generation Number and Objective Results Over 15 Runs

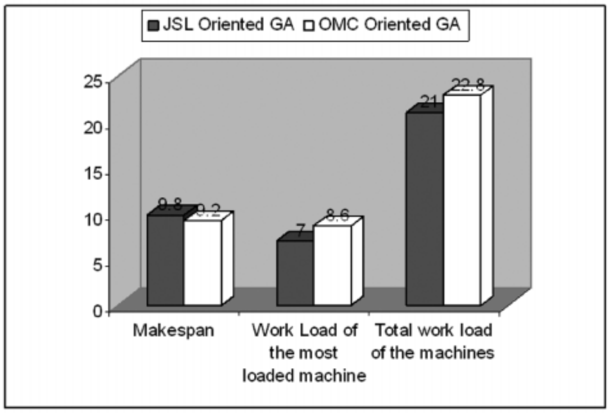

Fig. 3.2: Comparison between JSL Oriented GA and OMC Oriented GA for their Objectives to the First Input

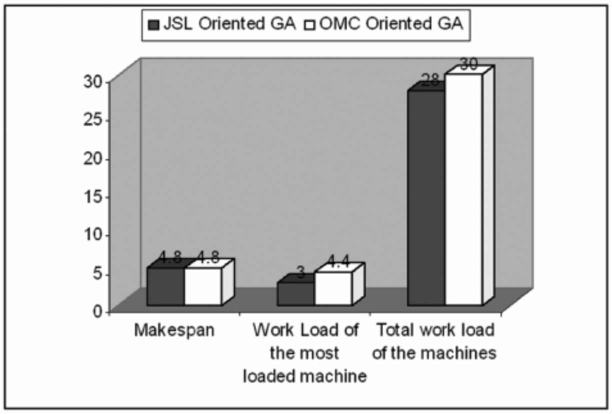

Fig 3.3: Comparison between JSL Oriented GA and OMC Oriented GA for their Objectives to the Second Input

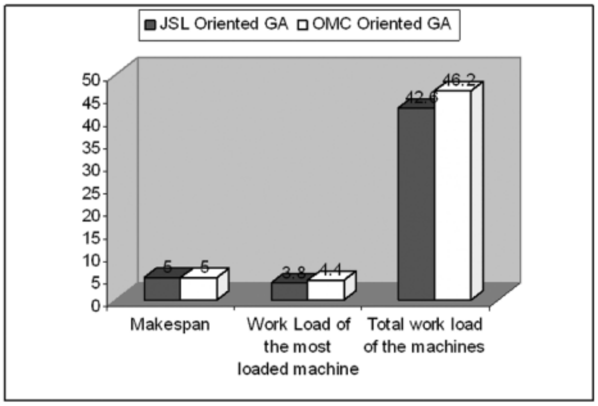

Fig. 3.4: Comparison between JSL Oriented GA and OMC Oriented GA for their Objectives to the Third Input

## 4. ANALYSES

### 4.1. Chromosome Representation

From two approaches it is concluded that in JSL and OMC oriented GAs have the sequence of operations as the first operations of the all job, second operations of all jobs and so on in the chromosome representation.

### 4.2. Evaluation Function

OMC oriented GA has mono criteria evaluation function. JSL GA has multi criteria evaluation functions. In the job-shop scheduling problem mono criteria evaluation function is better when only makespan objective is considered and multi criteria evaluation function is better when all three objectives are considered. In the multi criteria evaluation function weightage of the objective can be increased or decreased. Since all three objectives (makespan, workload of the most loaded machine and the total workload of machines) have been considered to solve the job-shop scheduling problem so multi criteria evaluation function is better.

### 4.3. Selection Process

In all two approaches roulette wheel and elitism selection processes have been used. In the roulette wheel selection best chromosome get more copies, the average stay even and the worst die off. Elitism first copies the best chromosome (or few best chromosomes) to the new population. The rest of the population is constructed by the roulette wheel selection.

### 4.4. Crossover

JSL oriented GA has two crossovers (1) sequencing crossover and (2) sequencing and assignment crossover. Sequencing crossover changes the sequence of jobs corresponding to the operations. Sequencing and assignment crossover changes the sequence of jobs corresponding to the operations and exchanges the machines assignment. OMC oriented GA has only one crossover, which exchanges the machine assignment. It guarantees the diversities of machines to be assigned. So crossovers of JSL oriented GA is better.

### 4.5. Mutation

In all two approaches two intelligent mutations have been used. Mutation I reduce the effective processing time, which helps to minimize the objectives the makespan and the total workload of the machine and mutation II balances the workload of the machines, which helps to minimize the objective the workload of the most loaded machine.

## 5. CONCLUSION

In this work, two approaches viz JSL oriented GA and OMC oriented GA have been used to solve job-shop scheduling problem. These two approaches have been compared. To be fair each run of JSL oriented GA and OMC oriented GA started with the same initial population. After each run of each approach we observed at which generation makespan, workload of the most loaded machine, total workload of the machine and all three objectives are minimum. We have calculated the average generation number at which objectives are minimum, average optimized makespan, average optimized workload of most loaded machine and average optimized total workload of the machines over 15 runs. Then we conclude that JSL oriented GA is better when all three objectives have equal weight age. And OMC oriented GA is better when makespan has most weight age.

## REFERENCES

[1] Y. Tsujimura, Y. Mafune, and M. Gen, “Effects of Symbolic Evolution in Genetic Algorithm for Job-Shop Scheduling”, 2001 IEEE.

[2] Y. Tsujimura, Y. Mafune, and M. Gen, “Introducing Co-evaluating and Sub-evolution Processes into Genetic Algorithm for Job-Shop Scheduling,” 2000 IEEE.

[3] Xiu Li, Wenhuang Liu, Shouju Ren and Xuerui Wang, “A Solution of Job-Shop Problem Based on Genetic Algorithm,” 2001 IEEE.

[4] H. Chen, J. Ihlow and C. Lehman, “A Genetic Algorithm for Flexible Job-Shop Scheduling,” 1999 IEEE.

[5] I. Kacem, S. Hammadi, and P. Borne, “Approach by Localization and Multiobjective Evolutionary Optimization for Flexible Job-Shop Scheduling Problem,” February 2002 IEEE.

[6] L. Davis “Job Shop Scheduling with Genetic Algorithm”, In J. J. Grefenstette, Editor, Proceedings of the International Conference, on Genetic Algorithms and their Applications, pp 136-140, San Mateo: Morgan Kaufmann, 1985.

[7] I. Kacem, S. Hammadi, P. Borne, “Direct Chromosomes Representation and Advanced Genetic Genetic Operators for Job-Shop Problems”, Proceedings of the International Conference on Computational Intelligence for Modeling, Control and Automation (CIMCA’01), 9-11 July 2001, Las Vegas, USA.

[8] I. Kacem, S. Hammadi, P. Borne, “Multi-objective Optimization for Flexible Job-Shop Scheduling Problem : Hybridization of Genetic Algorithms with Fuzzy Logic”, Proceedings of ifdicon’2001, European Workshop on Intelligent Forecasting, Diagnosis and Control, 24-28 June, 2001, Santorini, GREECE.

[9] Kacem, S. Hammadi, and P. Borne, “Approach by Localization and Genetic Manipulation Algorithm for flexible Job-Shop Scheduling Problem,” 2001 IEEE.

[10] M. T. Isaai and M. G. Singh, “An Object-oriented Constraint-based Heuristic for a Class of Passengers Train Scheduling Problems,” IEEE Trans. Syst., Man, Cybern. C, 30, pp. 12–21, Feb. 2000.

[11] K. Mesghouni, S. Hammadi, and P. Borne, “Evolution Programs for Job-shop Scheduling,” in Proc. IEEE Syst., Man, Cybern. Conf., 1, Orlando, FL, Oct. 12–15, 1997, pp. 720–725.

[12] L. Davis, “Job-shop Scheduling with Genetic Algorithms,” in Proc. Int. Conf. Genetic Algorithms and Their Applications. San Mateo, CA: Morgan Kaufmann, 1985, pp. 136–140.

[13] Ono, “A Genetic Algorithms for Job-shop Scheduling Problems using Job based Order Crossover,” in Proc. ICEC, 1996, pp. 574–552.

[14] I. Kacem, S. Hammadi, and P. Borne, “Pareto-optimality Approach for Flexible Job-shops Scheduling Problems: Hybridization of Evolutionary Algorithms and Fuzzy Logic,” J. Math. Comput. Simul., 2002.