Optimization of Job Shop Scheduling Using Genetic Algorithm

Sunil Kumar
Department of Mechanical Engineering,
FET, MRIU, Faridabad
sunilkumar.fet@mriu.edu.in

Dr. R.V. Singh
Department of Mechanical Engineering,
FET, MRIU, Faridabad
rvsingh.fet@mriu.ed.in

Abstract: *The Job-Shop Scheduling Problem (JSSP) is considered as one of the difficult combinatorial optimization problems and treated as a member of Non Polynomial problem class. The present work aims to develop a genetic algorithm based approach to solve the Job Shop scheduling problem. Operation based representation is used to decode the schedule in the algorithm. Two point crossover and flip inverse mutation is used in this algorithm. The algorithm is encoded and developed in MATLAB Software. The input parameters are operation time and operation sequence for each job in the machines provided. The Objective of this present work is to minimize the makespan. The experimental results show that the proposed GA, as compared to earlier GA, not only improves the quality of solutions but also reduces the overall computational time.*

Keywords: *Job Shop Scheduling, Genetic algorithm, Shortest Processing Time, and Makespan.*

I. INTRODUCTION

Scheduling problems exist almost everywhere in real industrial world situations. Scheduling is an optimization process intended to make the best possible use of the limited resources by making suitable allotment of the said resources over a period of time [2]. The job shop scheduling problem (JSSP) has been known as an extremely stubborn combinatorialoptimization problem since the 1950s [1].The problem so formulated is extremely difficult to solve, as it comprises several concurrent goals and several resources which must be allocated to lead to 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. The main problem is how to cope with local minima within a reasonable time.The Job Shop Scheduling Problem (JSSP) is constraint satisfaction problems that has interested to researches over the last years where the set of feasible solutions is discrete or can be reduced to a discrete one, and the goal is to find the best possible solution. Many researches involved job shop scheduling have been presented and various approaches have been implemented to solve this problem. Techniques such as Integer Programming, Branch and Bound [3], Taboo Search method [4], Genetic Algorithm [5] and Simulated Annealing [6] were widely used in recent years.GAs represent potential solutions by strings of symbols, or linear chromosome, and simulate the process of natural selection, crossover, and mutation among a population of chromosomes, as inspired by Darwinian evolution.

II. LITERATURE REVIEW

Many researchers have worked on Job Shop Scheduling using different techniques. A brief review of the researchers work on Job Shop Scheduling using different techniques along with their merits and demerits are mentioned below:

Maghfiroh et.al (2013) used genetic algorithm (GA) with some modifications to deal with problem of job shop scheduling. In job-shop scheduling problem

(JSSP), there were ‘k’ operations and ‘n’ jobs to be processed on m machines with a certain objective function to be minimized. Performance measures were mean flow time and make span. The result then was compared with dispatching rules such as longest processing time, shortest processing time and first come first serve. The proposed algorithm has been coded in Mat lab and investigated on a real case problem in which they found good solution value for the instances in every run (out of its 20 runs) compare to dispatching rules result. [7]

Li & Pan (2012) proposed a hybrid algorithm combining particle swarm optimization and Taboo search to solve the job shop scheduling problem with fuzzy processing time. The object was to minimize the maximum fuzzy completion time. TS-based local search approach was applied to the global best particle to conduct find-grained exploitation 8 benchmarks with different scales are conducted by the proposed algorithm. 1st four cases were a six-job–six-machine problem; the scale of following four problems was 10 jobs–10 machines. [8]

Bozejko & Makuchowski (2011)described a methodology of automatic genetic algorithm parameters adjustment dedicated to a job-shop problem with a no-wait constraint with a makespan criterion. The numerical results showed that in a given problem, the efficiency of an algorithm with auto-tuning has been placed at the level of an algorithm steered in a classical way with the best-fit steering parameters. More precisely, for all 40 tested instances, the HGA algorithm generates better solutions in 39 instances. [9]

Oliveira et.al (2010) presented a genetic algorithm for the Job Shop Scheduling Problem (JSSP). The genetic algorithm was based in random keys chromosome that was very easy to implement and allows using conventional genetic operators for combinatorial optimization problems. For each instance were made five runs considering the tail information on the initial population, and also were made five runs with an initial population generated randomly. In each run the algorithm performs 5000 iterations. [10]

III. PROBLEM STATEMENT

A ($n * m$) Job Shop Problem is defined by a specific number of jobs n, each consisting of an order of operations m**,** which are equal to the number of machines or resources specified in the problem. So a job, Ji is a predefined order of operations Oi = (O_{i1}, O_{i2}... O_{im}). Each operation O_{ij}has a processing timeT_{ij}.For the Job Shop Problem the following assumptions are considered:

- Each job consists of a finite number of operations.
- The processing time for each operation using a particular machine is defined.
- There is a pre-defined sequence of operations that has to be maintained to complete each job.
- A machine can process only one job at a time.
- Each job is performed on each machine only once.
- No machine can deal with more than one type of task.

Sr. No.	Author Name & Year	Population size	Selection Scheme	Crossover type & Crossover Prob.	Mutation type & Mutation prob.	Termination Criterion
1	Asadzadeh & Zamanifar (2010)	1000	———	0.95	0.1	1000
2	Gholami and Zandie (2009)	1000	———	0.9	0.05	200
3	GuJinwei et.al (2009)	50	Roulette wheel	Cycle crossover (0.8)	0.1	1000
4	Wang Yong Ming et.al (2008)	150	Random initialization	0.8	0.05	2,000
5	Mattfeld Dirk C. & Bierwirth Christian (2004)	100	Giffler and Thompson	Preservative (0.6)	Randomly chosen position (0.01)	50
6	Ombuki et.al (2004)	200	Random	0.9	0.1	550

- No machine can halt a job and start another job before finishing the previous one.
- The system cannot be interrupted until each operation of each job is finished.

Some Notations used in this paper are listed as follows:

- J= job (j=1, 2… n)
- M= machine (i=1, 2… m)
- TT= transfer time
- P= processing time
- W= waiting time
- C= completion time job

The problem has been taken from **Omar Mahanim, Baharum Adam, and Hasan Yahya Abu (2006)**.The objective function is to minimize makespan C_{max}.

IV. METHODOLOGY

Genetic algorithms are a family of computational models belonging to the class of evolutionary algorithms and a part of artificial intelligence. These algorithms encode a potential solution to a specific problem on a simple chromosome. It uses techniques inspired by natural evolution such as inheritance, mutation, selection and crossover. In 1975 John Holland [12] published book Adaptation in Natural and Artificial System. Most organisms evolve by means of two primary processes: natural selection and sexual reproduction. The first determines which members of population survive and reproduce, and the second ensures mixing and recombination among the genes of their offspring. Similar analogy is used in GA.

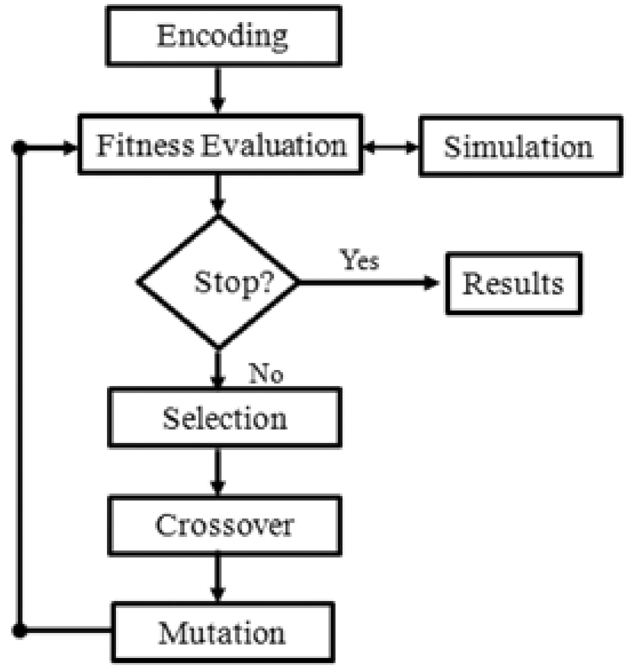

Fig. 1 : Flow Chart

(A) Population Generation and Representation: Before solving the JSSP, we need to describe a proper representation for the solution of the problem, namely a scheduling, which is used in the proposed algorithms. In this paper, we adopt an operation-based representation method.

(B) Evaluation of Chromosome's Fitness. Fitness function is defined of each chromosome so as to determine which with reproduce and survive into the next generation. It is relevant to the objective function to be optimized. The greater the fitness of a chromosome is, the greater the probability to survive.

Table 1: 5*5 benchmark Problem

Job No.	**Machine**	**Machine**	**Machine**	**Machine**	**Machine**
Job 1	3 (2)	1 (8)	2 (4)	4 (6)	5 (7)
Job 2	2 (6)	3 (5)	5 (4)	1 (3)	4 (2)
Job 3	1 (7)	5 (8)	4 (4)	3 (9)	2 (3)
Job 4	4 (4)	3 (5)	2 (5)	1 (4)	5 (3)
Job 5	5 (5)	3 (7)	1 (3)	2 (6)	4 (4)

(C) Selection: In this paper, **Tournament selection** is used to generate a new population for the next generation. A number of randomly selected individuals are chosen. A tournament is played among them based on the selection criteria. The winner of each tournament is selected for the next round and the final winner(s) of the tournament is selected for reproduction. A tournament can be performed between two parents or more than two.

(D) Crossover Operator. One of the important aspects of the technique involved in genetic algorithm is crossover. The crossover process is used to breed a pair of children chromosome from a pair of parent chromosomes using a crossover method. In this paper, **two point** crossover operator is used.

(E) Mutation: The mutation operation is critical to the success of theGA since it diversifies the search directions and avoids convergence to local optima. **Flip Bit** that simply inverts the value of the chosen gene (0 goes to 1 and 1 goes to 0) is used. This mutation operator can only be used for binary genes.

(F) Termination Criteria: The algorithm will be stopped if it reaches a specified maximum number of generations or if it reaches a specified maximum number of iterations without any improvement.

V. RESULTS

This section describes the computational tests which are used to evaluate the effectiveness and efficiency of the proposed algorithm in finding good quality schedules. In this process from one generation to the next generation, the cross over and mutation is repeated until the maximum number of generation is satisfied. The proposed algorithm is coded in MATLAB. The parameters used in this algorithm are shown in Table.2

Table 2: Evaluation Parameters

Parameters Used	Earlier GA	Existing GA
Population Size	500	200
Crossover Rate	0.95	0.8
Mutation Rate	0.15	0.2
No. of Iteration	200	1000
Makespan	55	34

The results obtained after implementing GA on 5 jobs and 5 machines problemare shown by Gantt chart in below Fig. 2 & 3:

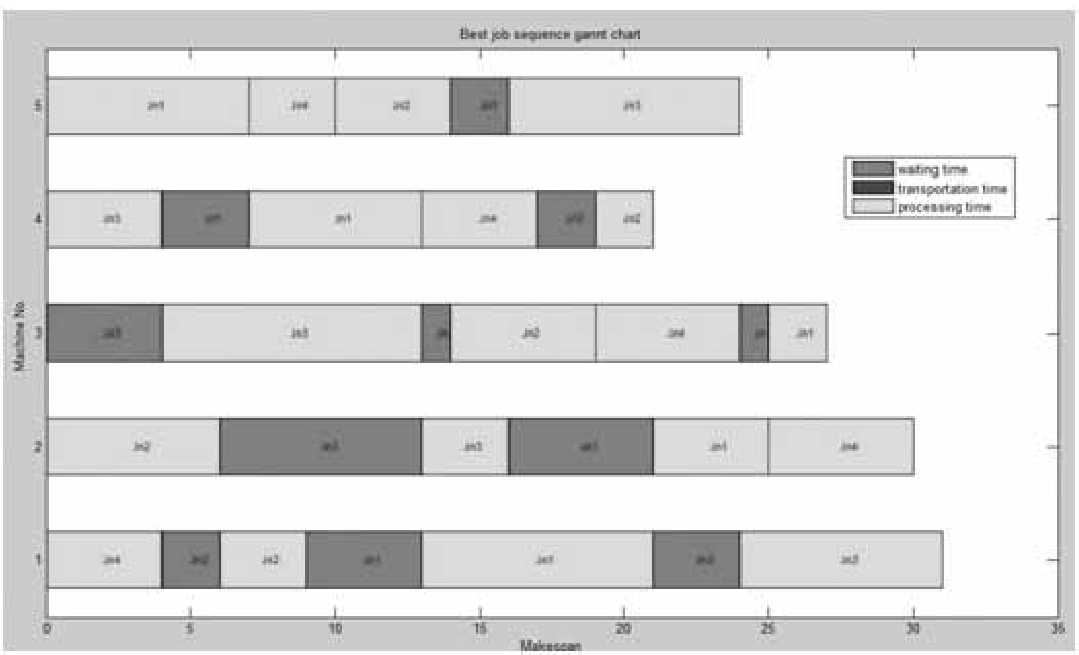

Fig. 2 : Gantt chart

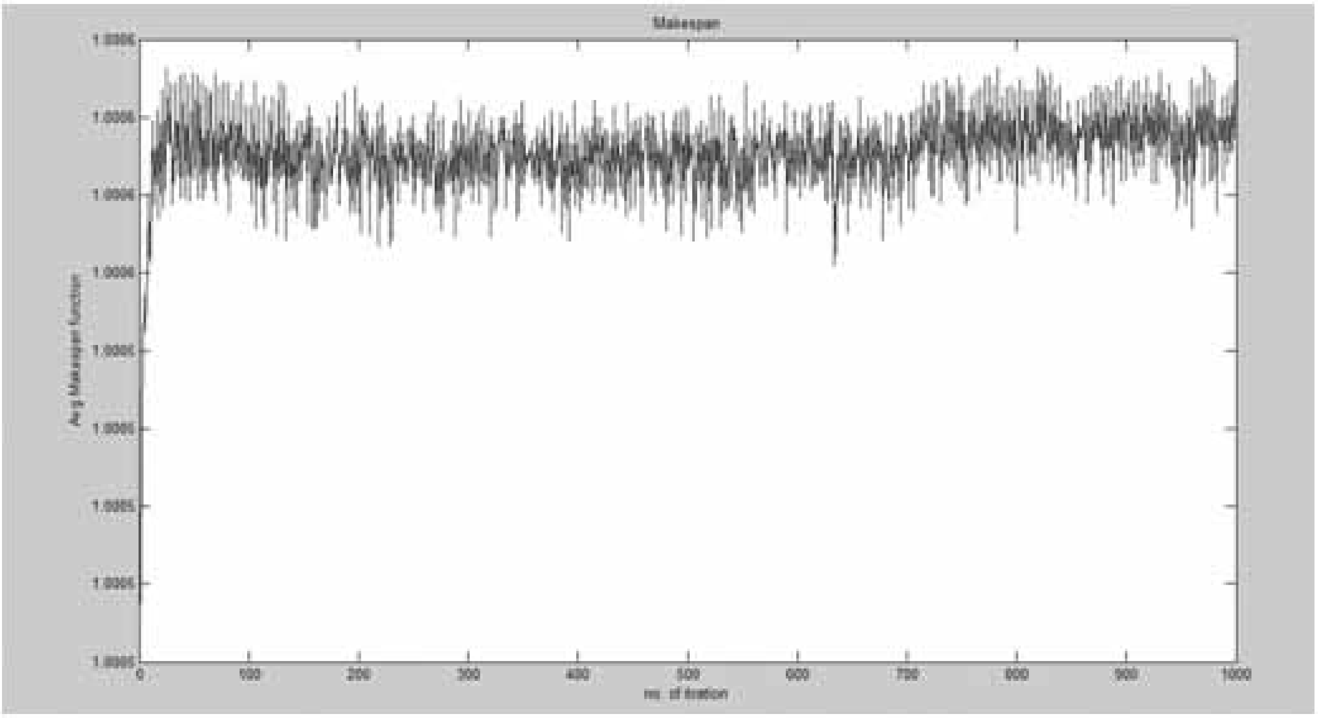

Fig. 3 : No. of Iteration vs Avg Makespan

VI. CONCLUSION

The optimum objective value obtained by GA for this problem is represented by Figure.2, i.e, makespan of given problem decreases from 55 units to 34 units. The computational result shows that GA can obtain better solution. GA result can obtain 38% improvement of the makespan. Further work includes considering other meta-heuristics for the job shop scheduling problem.

REFERENCES

[1]. Lenstra, J.K.; Kan, A.H.G.R.; Brucker, P (1977). “Complexity of machine scheduling problems”. Ann. Discret. Math.,1,343–362.

[2]. Baker KR (1974). “Introduction to Sequencing and Scheduling”. Wiley, New York.

[3]. Brucker P, Thiele O (1996). “A branch & bound method for the general shop problem with sequence dependent setup-times”. OR Spektrum;18: pp.145–161.

[4]. Liu, M., Dong, M. Y., & Wu, C. An iterative layered Taboo search algorithm for complex job shop scheduling problem.Chinese Journal of Electronics, 2005,14(3), pp.519–523.

[5]. Liu, T. K., Tsai, J. T., & Chou, J. H (2006). “Improved genetic algorithm for the job-shop scheduling problem”. International Journal of Advanced Manufacturing Technology, 27(9–10), pp. 1021– 1029.

[6]. Diaz-Santillan, E., & Malave, C. O (2004). “Simulated annealing for parallel machine scheduling with split jobs and sequence dependent set-ups”.International Journal of Industrial Engineering– Theory Applications and Practice, 11(1), pp. 43–53.

[7]. Asadzadeh Leila, and Zamanifar Kamran (2010). “An agent-based parallel approach for the job shop scheduling problem with genetic algorithms”, Mathematical and Computer Modeling 52.1957-1965.

[8]. Gholami M., and M. Zandieh(2009). “Integrating simulation and genetic algorithm to schedule a dynamic flexible job shop”.J IntellManuf 20:481–498.

[9]. GuJinwei, Gu Xing Sheng, and GuManzhan (2009). “A novel parallel quantam Genetic Algorithm for stochastic job shop scheduling.” J. Math .Anal Appl.355: 63-81.

[10]. Wang Yong Ming, Xiao Nan Feng, Yin Ho ng Li, Hu En Liang, Zhao Cheng Gui, and Jiang Yan Rong (2008). “A Two-stage genetic algorithm for large size job shop scheduling problems”. Int. J AdvManuf Techno 39:813–820.

[11]. Mattfeld Dirk C., and Bierwirth Christian (2004). “An efficient genetic algorithm for job shop scheduling with minimization of tardiness objectives”. European Journal of Operational Research 155:616–630.

[11]. Ombuki Beatrice, and Ventresca Mario (2004). “A Local Search Genetic Algorithms for the Job Shop Scheduling Problem”.Applied Intelligence 21, 99–109, Kluwer Academic Publishers. Manufactured in The United States.

[12]. Holland, J. H. (1975). “Adaptation in Natural and Artificial Systems”. University of Michigan Press. Second edition: MIT Press, 1992.