

ANALYSIS OF SELECTION SCHEMES FOR SOLVING JOB SHOP SCHEDULING PROBLEM USING GENETIC ALGORITHM

A.Ranjini¹, B.S.E.Zoraida²

¹ Research Scholar, Bharathidasan University, Tiruchirappalli, Tamilnadu, India, ranjini.anbu@gmail.com

² Assistant Professor, Bharathidasan University, Tiruchirappalli, Tamilnadu, India, b.s.e.zoraida@gmail.com

Abstract

Scheduling problems have the standard consideration in the field of manufacturing. Among the various types of scheduling problems, the job shop scheduling problem is one of the most interesting NP-hard problems. As the job shop scheduling is an optimization problem, Genetic algorithm was selected to solve it In this study. Selection scheme is one of the important operators of Genetic algorithm. The choice of selection method to be applied for solving problems has a wide role in the Genetic algorithm process. The speed of convergence towards the optimum solution for the chosen problem is largely determined by the selection mechanism used in the Genetic algorithm. Depending upon the selection scheme applied, the population fitness over the successive generations could be improved. There are various type of selection schemes in genetic algorithm are available, where each selection scheme has its own feasibility for solving a particular problem. In this study, the selection schemes namely Stochastic Universal Sampling (SUS), Roulette Wheel Selection (RWS), Rank Based Roulette Wheel Selection (RRWS) and Binary Tournament Selection (BTS) were chosen for implementation. The characteristics of chosen selection mechanisms of Genetic algorithm for solving the job shop scheduling problem were analyzed. The Genetic algorithm with four different selection schemes is tested on instances of 7 benchmark problems of different size. The result shows that the each of the four selection schemes of Genetic algorithm have been successfully applied to the job shop scheduling problems efficiently and the performance of Stochastic Universal Sampling selection method is better than all other four selection schemes.

Keywords: Genetic Algorithm, Makespan, Selection schemes

1. INTRODUCTION

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 [5]. There are several techniques that have been applied to solve the scheduling problems namely time tabling problem, flow shop scheduling, etc... One important problem of scheduling is the job-shop scheduling problem (JSSP). It is a famous problem in the field of industries especially, in manufacturing companies and transportation companies. Job shop scheduling is one of the combinatorial optimization problems, where the set of possible solutions is very large and the goal is to find the best possible solution. Traditional methods can be also be used to solve the job shop problem. But, the time complexity is there with them. This is because, when the problem size is small, the traditional methods for solving JSSP are able to explore the entire search state in order to obtain the optimal solution within reasonable time. But it is not the case for the most real world problems. Therefore, a non-traditional method known as "Genetic Algorithm" can be utilized in the scheduling of the manufacturing system. It is a very effective algorithm to search for solutions for an optimization problem. The solutions are usually optimum solutions or solutions near by the optimum value for the chosen problem.

A lot of work has been already carried out for solving the job shop problem. R.Thamilselvan and P.Balasubramanie [1] introduced three new crossover operators namely Order

Preserving Multipoint Crossover, Unordered Subsequence Exchange Crossover and Ordered Partially Mapped Crossover in the Genetic Algorithm for solving the JSSP. They analyzed the effectiveness of each of the three crossover operators by testing them on the known benchmark problems. From their results, it is proved that the Unordered Subsequence Exchange Crossover operator provides the remarkable performance than the other two operators. R.Sivaraj and Dr.T.Ravichandran [2] presented an overview of Genetic Algorithm and the various selection schemes used in the Genetic Algorithm. Tamer F.Abdelmaguid [3] provided a computational study to compare impact of chromosome representation in Genetic Algorithm for solving the JSSP. The performance of the Genetic Algorithm under six different chromosomes representations was presented in the paper. It is found that the machine-based representation is capable of achieving the lowest optimality gap, but with the highest computational time. Omar Al Jadaan, LakishmiRajamani[4] and C. R. Rao introduced a new selection method namely Rank Based Roulette Wheel selection in their work. The result shows that by introducing this selection scheme in the Genetic Algorithm the gain of diversity among population is increased and the uncertainty in selection process is decreased. Kumar Ritwik and Sankha Deb [5] developed a new encoding scheme for chromosome representation. The proposed encoding scheme is compared with other schemes on the basis of the Lamarkian property, complexity of decoder and memory requirement. The proposed scheme was compared with the existing operation based representation. When the number of operations is

small, the proposed scheme performs better than the operations based scheme. But, when the number of operations is increased, the proposed scheme tends to lose its effectiveness. S.M.Kamrul, RuhulSarker and David Cornforth [6] had presented a hybrid Genetic Algorithm for solving the JSSP. The proposed Genetic Algorithm includes a heuristic job ordering with the Genetic Algorithm and they incorporated a new, but simple local search method. NorainiMohdRazali and John Geraghty [7] presented the comparison of Genetic Algorithm performance using different parent selection strategy in solving travelling salesman problem. Therefore, the selection strategy has notable impact for the problem using Genetic Algorithm. This factor is analyzed in this paper.

2. PROBLEM DESCRIPTION

The classical job-shop scheduling problem can be formulated as follows [1-5]: A set of n jobs and a set of m machines are given. The n jobs are to be scheduled on the m machines, where each job is divided into a set of operations. Each operation is needed to be processed by the required machine with the fixed processing time. There are several constraints on jobs and machines:

1. Every job has to be processed on all machines.
2. Each job must visit each machine exactly once.
3. Each machine can process only one job at a time.
4. The processing order of the set of operations for each job is pre-specified, which is known as "precedence constraint" or "technology constraint".
5. There are no precedence constraints among the operations of different jobs.
6. All operations are not preemptive. That is, each operation needs to be processed on the required machine without interruption for the given period of time.
7. The operations of the same job cannot be processed concurrently.
8. We assume that there is no machine failure.
9. Neither release times nor due dates are specified.

The maximum completion time of all the jobs is known as 'Makespan'. The execution sequence of all operations of all jobs on the given machines is referred as a schedule. A schedule is said to be feasible schedule only if it satisfies all the above stated constraints. The objective of the JSSP is to find a feasible schedule of minimum length, that is, the schedule must minimize the makespan.

3. GENETIC ALGORITHM

Genetic Algorithms are bio-inspired search algorithms based on the evolutionary ideas of natural selection and genetics that are used successfully to solve problems in many different disciplines. Genetic Algorithm uses the technique that resembles natural selection in the biological process. Genetic Algorithm was developed by John Holland, his colleagues, and his students at the University of Michigan in the 1960s and the 1970s. Due to the robustness of Genetic Algorithms on problems of high complexity, it has an increasing number

of applications in the fields of artificial intelligence, numeric and combinatorial optimization, business, management, medicine, computer science, engineering etc.

4. GENETIC ALGORITHM FOR JSSP

4.1 Operation based representation

In the scheduling problems, the popular representation is operation-based representation for the chromosome. In this study also, the operation based representation is adopted for implementation. This representation encodes a schedule as a sequence of operations where each gene stands for one operation. All operations of the same job are represented by the same symbol and they are interpreted according to the order of their occurrence in the chromosome sequence. Using the representation, each job number occurs m times in the chromosome. By scanning the chromosome from left to right, the k -th occurrence of a job number refers to the k -th operation in the technological sequence of this job.

2	2	1	3	1	2	2	3	1	3	3	2	1	3	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Fig-1. Example for chromosome representation

As in Fig- 1, an example chromosome for 3 X 5 JSSP is given as [2 2 1 3 1 2 2 3 1 3 3 1] for the three jobs and five machines problem. As each job consists of five operations, the job number occurs exactly five times in the chromosome. The fourth gene represents the first operation of job 3 because number 3 has been occurred at the first time. Similarly, the sixth gene in the chromosome implies third operation of job 2.

4.2 Permutation encoding

There are many chromosome encoding methods are available. Since the permutation encoding is only for ordering problem, this encoding is applied in this paper.

4.3 Initial population

In this paper, the initial population is created by means of random permutation of the operation sequence. It is proved that the initial solution methods affect the speed of convergence solution and so the better initial solutions might provide better results.

4.4 Fitness calculation

The fitness function for the JSSP can be defined as fitness = $1/(\text{makespan})$.

4.5 Selection

The selection strategy determines which of the chromosomes in the current generation will be used to reproduce offspring with the hope that the next generation will have higher fitness. Different selection strategies have different methods of calculating selection probability. All the differing selection techniques develop solutions based on the principle of survival of the fittest. In this paper, selection schemes namely Stochastic Universal Sampling, Roulette Wheel Selection,

Rank Based Roulette wheel selection and Binary Tournament selection were implemented.

4.6 Unordered subsequence exchange crossover

The Unordered Subsequence Exchange Crossover (USXX) is adopted in this work that children inherit subsequences on each machine as far as possible from parents [1]. Fig- 2 shows the method for performing the Unordered Subsequence Exchange Crossover and the procedure to do it is as follows.

- Step 1: Choose two parent individual randomly, namely Parent1 and Parent2.
- Step 2: Select random subset of operations (genes) from Parent1 and copy it into Child1 with the same context dependent.
- Step 3: Starting from the first crossover point from Parent1, look for elements within the subsequence in Parent2 and remove those from Parent2.
- Step 4: The remaining operations of Parent2 are copied into Child1 so as to maintain their relative ordering.
- Step 5: Change the Parents and go to Step 1 to generate Child2 in the same manner.

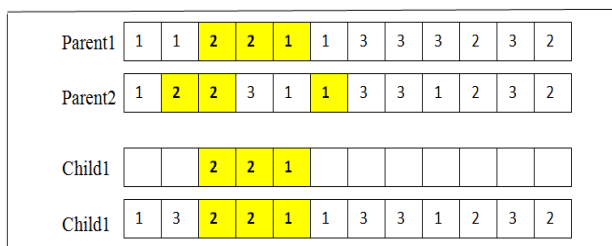


Fig-2.Unordered Subsequence exchange Crossover for 3 X 4 problem

4.7 Swap mutation

In this study, the swap mutation is applied. It randomly takes two positions in a chromosome and then the alleles in those positions are swapped. Fig -3 shows the procedure for performing the swap mutation for an example chromosome for 3 X 4 JSSP.

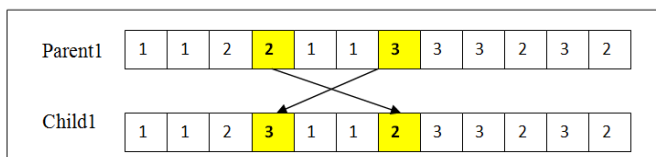


Fig-3. Swap Mutation for 3 X 4 problem

4.8 Procedure For Genetic Algorithm For JSSP

A simple and basic Genetic Algorithm cycle for solving the JSSP is shown in Fig- 4 as follows:

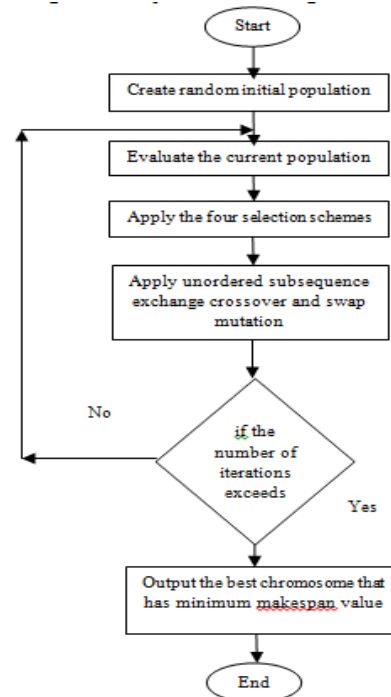


Fig-4. Procedure for Genetic Algorithm for JSSP

5. RESULTS AND DISCUSSION

The implementation work is carried out using Matlab software. In order to analyze the effectiveness of various selection schemes in this paper, the following factors are taken to be measured for each of the four selection schemes.

- Frequency of occurrence of optimum solution
- Fitness Evolution History and
- Quality of solution

Eight well known benchmark problems namely LA1, LA5, LA6, LA7, LA8, LA11, FT10 and LA13 are tested to measure these factors.

Frequency of occurrence of optimum solution

The frequency of occurrence of optimum solution refers to the number of occurrences of known optimum solution out of a number of trial runs. The result obtained for this factor is given in Table 1.

Table 1
Frequency Of Occurrence Of Optimum Solution

S. NO	PROBLEM NAME	FREQUENCY OF OCCURRENCE OF OPTIMUM SOLUTION			
		STOCHASTIC UNIVERSAL SAMPLING	RANK BASED ROULETTE WHEEL	ROULETTE WHEEL	BINARY TOURNAMENT
1	LA1	3	0	1	2
2	LA5	5	4	4	3
3	LA6	5	3	4	2
4	LA7	5	3	0	1
5	LA8	7	7	7	2
6	LA11	7	2	5	6
7	LA13	4	2	4	3

Table 1: Frequency Of Occurrence Of Optimum Solution

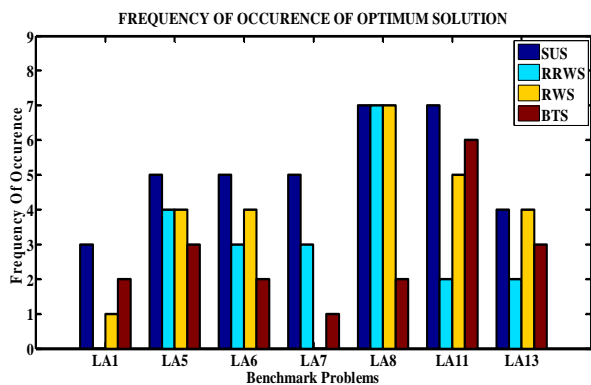


Fig -5. Frequency of occurrence of optimum solution

The Fig -5 is drawn using the data given in the Table 1. From this figure, it is known that the Stochastic Universal Selection(SUS) scheme has the high frequency of occurrence of optimum solution for the Rank Based Roulette Wheel Selection (RBWS), Roulette Wheel Selection (RWS) and Binary Tournament Selection (BTS). The empty spaces between the bars in the bar stack of a problem shows the zero frequency.

Fitness evolution history

The fitness evolution history shows the fitness value calculated for each iteration of the Genetic algorithm cycle. It is also used to refer the number of iterations that take to reach the optimum solution.

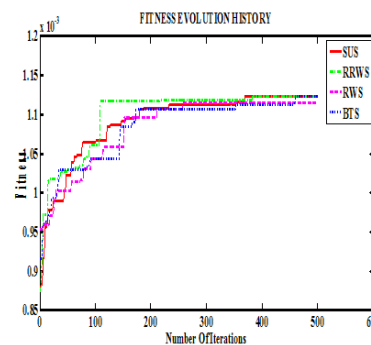


Fig-8.Fitness Evolution History For Problem LA7

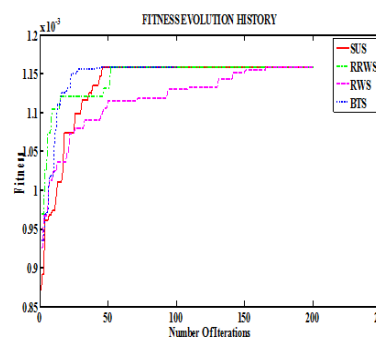


Fig-9.Fitness Evolution History For Problem LA8

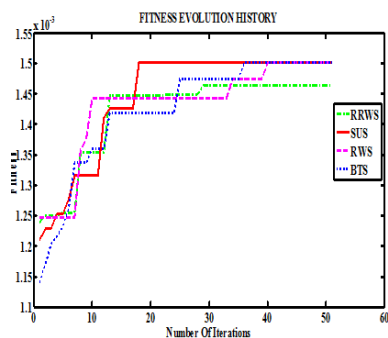


Fig-6. Fitness Evolution History For Problem LA1

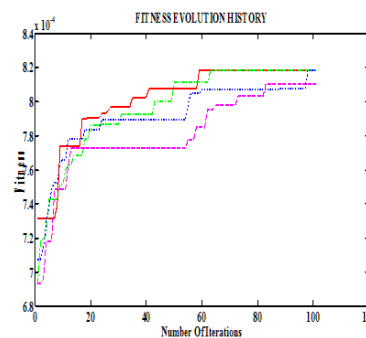


Fig-10. Fitness Evolution History For Problem LA11

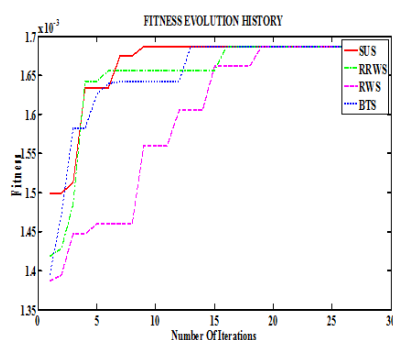


Fig-7. Fitness Evolution History For Problem LA5

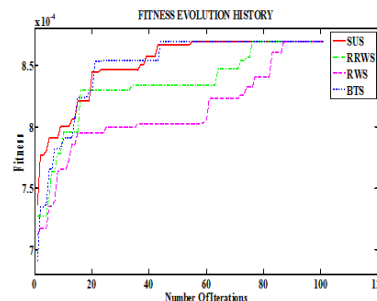


Fig-11. Fitness Evolution History For Problem LA13

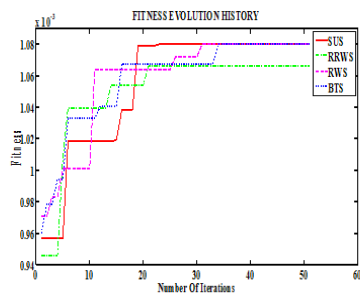


Fig-12. Fitness Evolution History For Problem LA6

From the Fig - 6, 7, 8, 9, 10, 11 and 12 it is known that the SUS scheme has reached the optimum solution with the less number of iteration when compared to all other three schemes.

Quality of Solution

The quality of solution is used to measure the deviation of obtained solution from the optimum solution of the problem. The result obtained for this factor is given in Table 6 and the Fig -13 is drawn using the data given in the Table 2.

Table 2
Quality Of Solution

S. N O	PROBLE M NAME	OPTIMUM VALUE	OBTAINED VALUES				DIEVIATION FROM OPTIMUM VALUE IN			
			SUS	RRWS	RWS	BTS	SUS	RRWS	RWS	BTS
1	LA1	666	666	683	666	666	0	2.55	0	0
2	LA5	593	593	593	593	593	0	0	0	0
3	LA6	926	926	926	926	926	0	0	0	0
4	LA7	890	890	890	897	890	0	0	0.78	0
5	LA8	863	863	863	863	863	0	0	0	0
6	LA11	1222	1222	1222	1222	1222	0	0	0	0
7	LA13	1150	1150	1150	1150	1150	0	0	0	0
8	FT10	930	979	1008	991	999	5.27	8.39	6.56	7.42
Total							5.27	10.94	7.34	7.42

Table 2: Quality of Solution

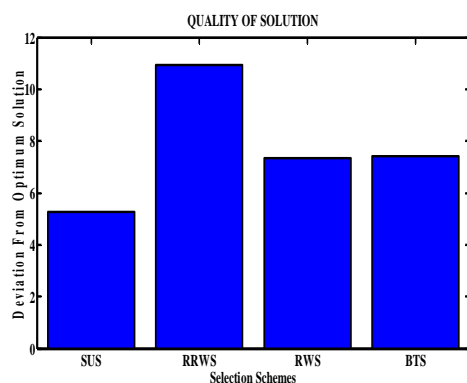


Fig -13: Deviation from optimum solution

From the Fig -13, it is shown that SUS scheme provides the minimum deviation from the optimum solution given than other three schemes.

CONCLUSION

From the experimental result, it is concluded that out of the four selection schemes for the tested benchmark problem, the Stochastic Universal Sampling selection scheme gave the high frequency of occurrence of optimum solution for all

problems. The average makespan value and the deviation from the optimum solution given by the scheme are also lesser it than the other selection schemes. Thus, the Stochastic Universal Sampling selection is well suited for solving the JSSP using Genetic Algorithm.

FUTURE WORK

The future work will concentrate on making the various parameters of Genetic Algorithm namely the population size, crossover probability and mutation probability can be made as dynamic. Some other techniques namely Simulated Annealing, Neighborhood search can also be incorporated as a local search technique in the Genetic Algorithm to improve its performance. The work can be extended for flexible job shop scheduling environment also.

ACKNOWLEDGEMENT

The authors are thankful to the Department of Computer Science, Engineering and Technology, Bharathidasan University, Trichy and all the people who help to make this study as a successful one.

REFERENCES

- [1] R.Thamilselvan and P.Balasubramanie, "Analysis of Various Alternate Crossover Strategies for Genetic Algorithm to Solve Job Shop Scheduling Problems" European Journal of Scientific Research ISSN 1450-216X Vol.64 No.4 (2011), pp. 538-554© EuroJournals Publishing, Inc. 2011
- [2] R.Sivaraj and Dr.T.Ravichandran, "A Review Of Selection Methods In Genetic Algorithm" International Journal of Engineering Science and Technology (IJEST) ISSN: 0975-5462 Vol. 3 No. 5 May 2011
- [3] Tamer F. Abdelmaguid, "Representations in Genetic Algorithm for the Job Shop Scheduling Problem: A Computational Study", Journal of Theoretical and Applied Information Technology(2005-2008)
- [4]Omar Al Jadaan, Lakishmi Rajamani, C. R. Rao," Improved Selection Operator For Ga", International Journal of computer science and technology (2006)
- [5] Kumar Ritwik and Sankha Deb, "A Genetic Algorithm-Based Approach For Optimization Of Scheduling In Job Shop Environment" Journal of Advanced Manufacturing Systems Vol. 10, No. 2 (2011) 223–240c_World Scientific Publishing Company DOI: 10.1142/S0219686711002235
- [6] S. M. Kamrul Hasan, *Student Member, IEEE*, Ruhul Sarker, *Member, IEEE*, and David Cornforth, "Hybrid Genetic Algorithm for Solving Job-Shop Scheduling Problem", Proceedings of the World Congress on Engineering 2011 Vol II WCE 2011, July 6 - 8, 2011, London, U.K. ISBN: 978-988-19251-4-5 ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)
- [7] Sivanandam S.N. and Deepa S.N. 2008. *Introduction to Genetic Algorithms*, Springer, Berlin, New York.
- [8] Goldberg D.E. (2003), "*Genetic Algorithms in Search, Optimization, and Machine Learning*" (Addison Wesley, Reading g, MA.)
- [9] <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/jobshopinfo.html>