Optimization of Job Shop Scheduling with Transportation Time Using Genetic Algorithm

Sunil Kumar
MED, FET, MRIU
E-mail: Sunilkumar.fet@mriu.edu.in

Rakesh Kr. Phanden
MED, MMU

R.V. Singh
MED, FET, MRIU
E-mail: rvsingh.fet@mriu.edu.in

Abstract: Efficiency in Job Shop Scheduling plays an important role when a large number of jobs and machines are considered. The high complexity of the problem makes it hard to find the optimal solution within reasonable time in most cases. This work deals with the Job Shop Scheduling (JSS) using Genetic Algorithm (GA). For a job-shop scheduling, 'n' number of jobs on 'm' number of machines processed through an assured objective function to be minimized.Objective of this present work is to minimize the makespan (The total time between the starting of the first operation and the ending of the last operation, is termed as the makespan). The input parameters are operation time and operation sequence for each job in the machines provided.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 proposed genetic algorithm with certain operating parameters is applied to the two case studies taken from literature. The results obtained from our study have shown that the proposed algorithm can be used as a new alternative solution technique for finding good solutions to the complex Job Shop Scheduling problems with shortest processing time and transportation time.

Keywords: Job Shop Scheduling, Genetic Algorithm,Transportation Time, Makespan

I. INTRODUCTION

In today's marketplace, a high level of delivery performance has increasingly become a tool to secure competitive advantages. This makes scheduling play a major role in manufacturing processes and in overall supply chain management.Scheduling is the process of deciding how to commit resources between varieties of possible tasks. Scheduling is an important tool for manufacturing and engineering, where it can have a major impact on the productivity of a process. In manufacturing, the purpose of scheduling is to minimize the production time and costs, by telling a production facility when to make, with which staff, and on which equipment. Production scheduling aims to maximize the efficiency of the operation and reduce costs.

In Job shop scheduling problems, there is also a set of 'n' number of jobs and 'm' number of machines. Each job has a number of operations and particular sequence of machines which it has to follow. "If there are 'n' jobs and 'm' machines the number of theoretically possible solutions is equal to $(n!)^m$. Among these solutions an optimal solution, for a certain measure of performance, can be found after checking all the possible alternatives. But the checking of all the possible alternatives can only be possible in small size problems. Because of its large solution space JSSP is considered to be comparatively one of the hardest Non- Polynomial problemsto solve.Generally, it is extremely difficult to solve this type of problems in their general form, 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 minimize the time required to complete the entire process being scheduled. A vast amount of research has been performed in this particular area to effectively schedule jobs for various objectives such as to minimize makespan, tardiness, mean flow time and to solve interruption on the shop floor like machine overloads, breakdowns and rush orders. Various techniques such as Branch and Bound, Integer linear Programming, Taboo Search, Genetic Algorithm and Simulated Annealing have been implemented to solve job shop scheduling problem. Even no algorithm is successful to solve such a problem optimally, up to today.The objective considered in this work is minimization of makespan which includes the processing times of the jobs. The genetic algorithm has been used to find the optimal schedule with minimum makespan and it can be used in System Identification, Control Systems Engineering, Robotics, Pattern Recognition, Engineering Designs and Planning and Scheduling etc.

II. LITERATURE REVIEW

A majority of articles on the topic Job Shop Scheduling focus on the relationship between Genetic Algorithm and performance. The purpose of scheduling is to minimize the production time and costs, by telling a production facility when to make, with which staff, and on which equipment. 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 is mentioned below:

Wang et.al (2013) proposed a novel genetic algorithm for flexible job shop scheduling problems with machine disruptions. They compared their novel approach to two benchmark algorithms: a right-shifting re-scheduler and a pre-scheduler. A right-shifting re-scheduler repairs schedules by delaying affected operations until the disruption is over. A pre-scheduler works on each disruption scenario separately, treating disruptions like prescheduled downtime. The genetic algorithms were parameterized with a population of 100 chromosomes, crossover probability of 0.7 and mutation probability of 0.1. Each algorithm was run 10 times for each instance.Future work includes applying the optimal computing budget allocation method in order to compare other parameter-dependent heuristic algorithms, such as simulated annealing, artificial neutral networks, and so on, and using the two-stage GA for the solution of other large size combinatorial optimization problems.

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 6 job–6 machine problem; the scale of following four problems was 10 jobs–10 machines.

Teekeng & Thammano (2012) proposed a modified version of the genetic algorithm for flexible job-shop scheduling problems (FJSP). The genetic algorithm (GA), a class of stochastic search algorithms, is very effective at finding optimal solutions to a wide variety of problems. The proposed modified GA consists of 1) an effective selection method called fuzzy roulette wheel selection, 2) a new crossover operator that uses a hierarchical clustering concept to cluster the population in each generation, and 3) a new mutation operator that helps in maintaining population diversity and overcoming premature convergence. The objective of this research was to find a schedule that minimizes the makespan of the FJSP. The genetic algorithms were parameterized with a population of 200 chromosomes, crossover probability of 0.9 and mutation probability of 0.3.

Zhang et. al (2011) presented **a**n effective genetic algorithm for the flexible job-shop scheduling problem. The parameters were tournament selection method; maximum generation number was 100 with a population size equal to 50-500 individuals, crossover 0.5 and mutation 0.1. The computational results showed that the proposed effective genetic algorithm leads to the same level or even better results in computational time and quality compared with other genetic algorithms.

Asadzadeh & Zamanifar(2010) proposed an agent-based parallel approach (PA) for the problem in which creating the initial population and parallelizing the genetic algorithm were carried out in an agent-based manner. Benchmark instances were used to investigate the performance of the proposed approach. They set the parameter values for both the serial and the parallel genetic algorithms as population size 1000, crossover rate 0.95, mutation rate 0.1, and generation spans 1000. In the parallel approach, communication between sub-populations of various PAs was carried out by exchanging migrants in the migration phase.

Liang et. al (2010) proposed a promising genetic algorithm with penalty function for the job shop scheduling problems. The proposed algorithm effectively exploits the capabilities of distributed and parallel computing of swarm intelligence approaches and effectively makes use of the famous scheme theorem and the building block hypothesis of Holland. The algorithm has been tested on a set of 43 benchmark instances. Parameters were population size 150, random selection method, hyper-mutation and termination criteria 1000. Simulation results were compared with those obtained using other competitive approaches. The results indicate the successful incorporation of the proposed operators. In the future work, they would aim to extend the proposed algorithm to be applied to more practical and integrated problems.

Zhang et. al (2010) proposed an effective genetic algorithm for the flexible job-shop scheduling problem to minimize makespan time. In the proposed algorithm, Global Selection (GS) and Local Selection (LS) were designed to generate high-quality initial population in the initialization stage. An improved chromosome representation was used to conveniently represent a solution of the FJSP, and different strategies for crossover LOX operator 0.8 and swap mutation 0.1 operators were adopted. They test problem with 15 jobs and 8 machines, ran algorithm 5 times on the same instance& the population size of 200.

Defersha & Chen (2009) presented a mathematical model for a flexible job-shop scheduling problem incorporating sequence-dependent setup time, attached or detached setup time, machine release dates, and time lag requirements. In order to efficiently solve the developed model, they proposed a parallel genetic algorithm (PGA) that runs on a parallel computing platform. The parameters were population size 500-5000 and termination criterion 50000. The results obtained using the PGA were very promising and encouraging compared to those obtained using the SGA. In their future research, they plan to extend the model and the solution procedure to consider multiple objectives such as workload balancing, due dates, and mean flow-time requirements, and others factors such as capacitated buffer and transportation time.

Gholami & Zandie (2009) described how they can integrate simulation into genetic algorithm to the dynamic scheduling of a flexible job shop with machines that suffer stochastic breakdowns. The objectives were the minimization of two criteria, expected makespan and expected mean tardiness. Jobs ranging from 10 to 20, number of machines ranging from 4 to 15 and number of operations for each job ranging from 5 to 15. Population size: 1000, crossover probability: 0.90, mutation probability: 0.05 and number of generations: 200. The results obtained reveal that the relative performance of the algorithm for both abovementioned objectives has been affected by changing the levels of the breakdown parameters.

Rossi & Boschi (2009) presented an advanced software system for solving the flexible manufacturing systems (FMS) scheduling in a job-shop environment with routing flexibility, where the assignment of operations to identical parallel machines has to be managed, in addition to the traditional sequencing problem. Two of the most promising heuristics from nature for a wide class of combinatorial optimization problems, genetic algorithms and ant colony optimization (ACO), share data structures and co-evolve in parallel in order to improve the performance of the constituent algorithms. A modular approach has been also adopted in order to obtain an easy scalable parallel evolutionary-ant colony framework. 15 Lawrence instances with duplicate and triplicate resources and jobs were adopted for 30 instances &solve to optimality 10 duplicate & 2 triplicate instances and achieve the best results in 23 of the 30 instances.

III. PROBLEM DEFINITION

Literature review reveals that many researchers have been done into the field of evolutionary computations or meta-heuristic techniques particularly with Genetic Algorithm. No attempts so far have been made with Genetic Algorithm in addition with Transportation Time and Shortest Processing Time as dispatching rule for job-shop scheduling problem. The most basic version of job shop can be defined as: Job set, J ={J1, J2… J_j} | j = 1, 2, 3… n and Machine set, M = {M1, M2… M_i}| i = 1, 2, 3… m and Operations set, O = {O1, O2… Oo} | o = 1, 2, 3… k. Job J_i consists of O_i operations and each operation is associated with a set of processing machines and a set of processing time. Processing time for each operation is given by, P_{ij} = {P11,P12,………,P_{ij}},| i = 1,2,3,……,n ; j =1,2,3,……,m. When a job has finished an operation and needs to move to another machine, a transport resource is required. Some assumptions used in this paper are:

- All jobs are available for processing at time zero.

Sr. No.	Author Name &Year	Population Size	Selection Scheme	Crossover Type & Crossover Probability	Mutation Type & Mutation probability	Termination Criterion
1.	Meilinda F. et.al (2013)	45	Roulette wheel	2 point Random (0.55)	Swapping (0.13)	2000
2.	Mendes (2013)	5 (no. of Activties)	Roulette Wheel	1 point	1 point (0.001)	10000
3.	Ren et.al (2013)	100	--	Crossover probability (0.7)	Mutation probability (0.1)	10 independent
4.	Yew Wong (2011)	--	Tournament Selection	Random (0.9)	Random (0.05)	5000
5.	Jinwei et.al (2009)	50	Roulette wheel	Cycle crossover (0.8)	0.1	1000
6.	Wang et.al (2008)	150	Random initialization	0.8	0.05	2,000
7.	Zhang et.al (2008)	200	--	precedence operation crossover (POX)	Insertion mutation and Inversion mutation 0.8	50–80
8.	Gao et.al (2006)	1000	Ranking selection	one-cut probability (0.30)	Allele-based mutation probability (0.10)	200
9.	Omar et.al (2006)	10	Random Selection	0.5	0.5	100
10.	Ombuki et.al (2004)	200	Random	0.9	0.1	550

- The transportation time between operations will be occurred whenever there is a machine changes for each job.
- There is only one machine of each type in the shop.
- Processing times for all jobs are known and constant.
- Each machine can perform only one operation at a time on any job.
- An operation of a job can be performed by only one machine.
- An operation of a job cannot be performed until its preceding operations are completed.
- Each machine is continuously available for production.
- There are no limiting resources other than machines/workstations.

Some Notations used in this paper are listed as follows:

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

Case Study: The problem has been taken from Ombuki Beatrice M. and Mario Ventresca, Applied Intelligence 21, 99–109, 2004.Table.1 shows a problem of 6 jobs and 6 machines with their sequence and processing time. The objective function is to minimize makespan C_{max}.

Table 1: 6*6 problem instance

Job	m,t	m,t	m,t	m,t	m,t	m,t
Job 1	3,1	1,3	2,6	4,7	6,3	5,6
Job 2	2,8	3,5	5,10	6,10	1,10	4,4
Job 3	3,5	4,4	6,8	1,9	2,1	5,7
Job 4	2,5	1,5	3,5	4,3	5,8	6,9
Job 5	3,9	2,3	5,5	6,4	1,3	4,1
Job 6	2,3	4,3	6,9	1,10	5,4	3,1

IV. METHODOLOGY

The father of original Genetic Algorithm was John Holland who invented it in early 1970. Genetic algorithms are adaptive methods, which may be used to solve search and optimization problem. They are based on the genetic process of biological organisms. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. Over many generations, natural populations evolve according to the principles of natural selection, i.e. survival of the fittest, first clearly stated by Charles Darwin in The Origin of Species. By mimicking this process, genetic algorithms are able to evolve solutions to real world problems, if they have been suitably encoded. Before a genetic algorithm can be run, a suitable encoding or representation for the problem must be devised. A fitness function is also required, which assigns a figure of merit to each encoded solution.The flow graph of implementation of Genetic Algorithm is shown in Fig. 1:

(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. Forexample, suppose a chromosome is given as [123456,234561,345612, 456123, 561234, 612345] problem.

(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.

(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.

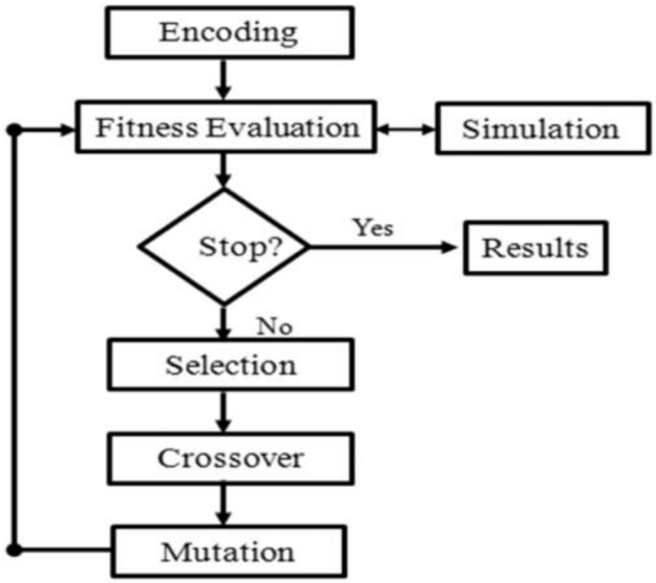

Fig.1: Flow Graph Genetic Algorithm

V. RESULT

This section describes the computational tests which are used to evaluate the effectiveness and efficiency of the proposed algorithm to find good quality schedules. The maximum number of generation (G) is selected as the stopping criteria. 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	GA in Case Study
Population Size	800
Crossover Rate	0.8
Mutation Rate	0.2
No. of Iteration	200
Makespan	47

The results obtained after implementing GA on 6 jobs and 6 machines problemare shown by Gantt chart in below Fig:

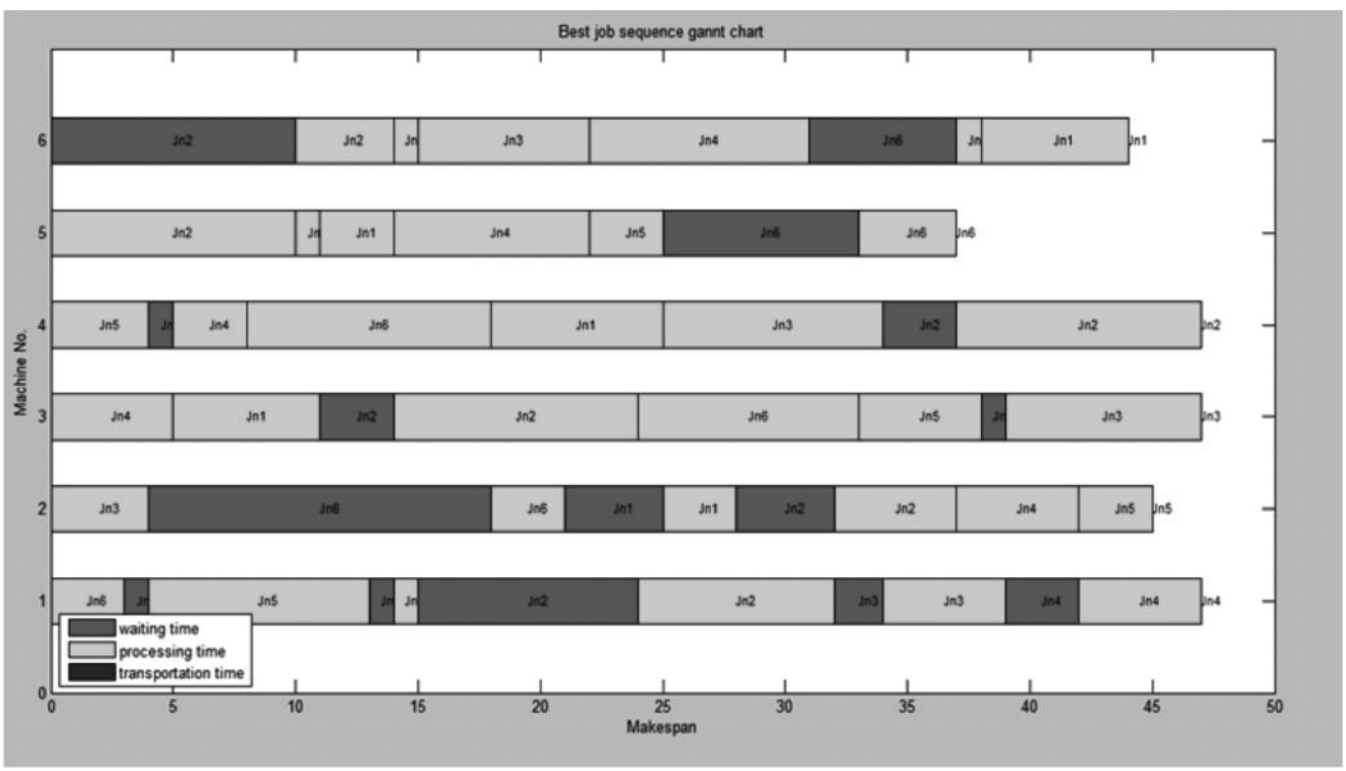

Fig. 2: Gantt chart

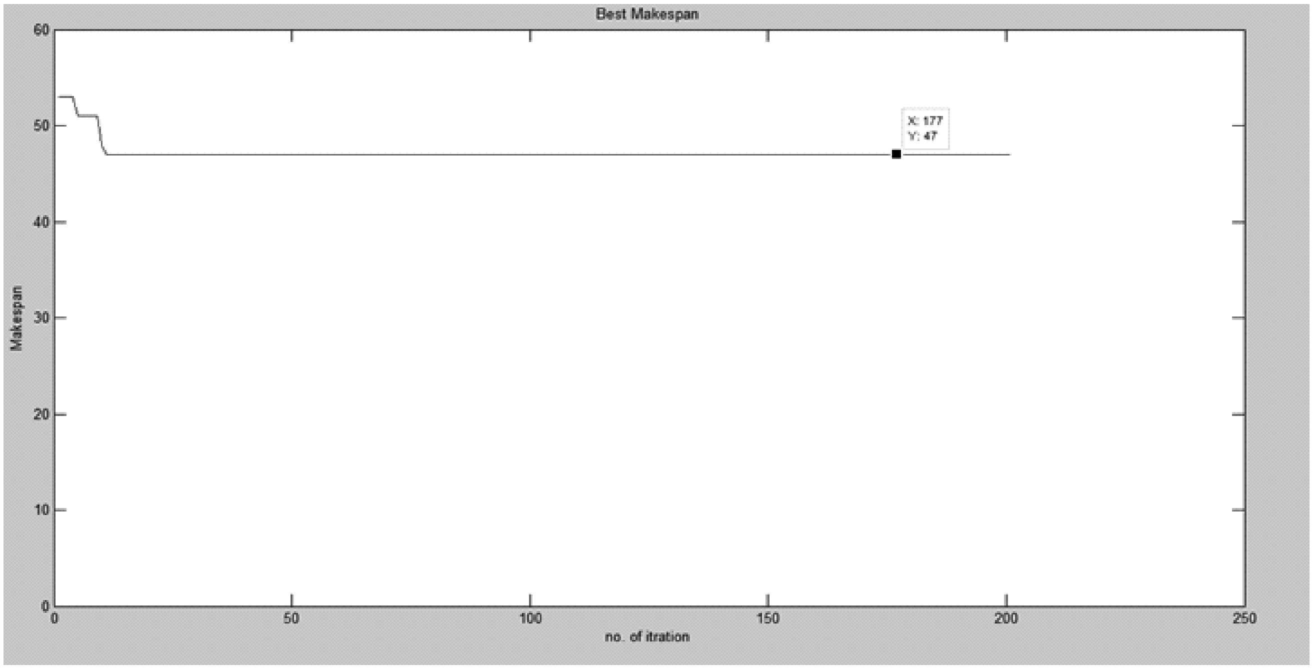

Fig. 3: Makespan vs Iteration

Further, if we use Transportation Time (The transportation time between operations will be occurred whenever there is a machine changes for each job) with the above problem then the solution is:

After including Transportation to the above problem the makespan is 63 units of time.

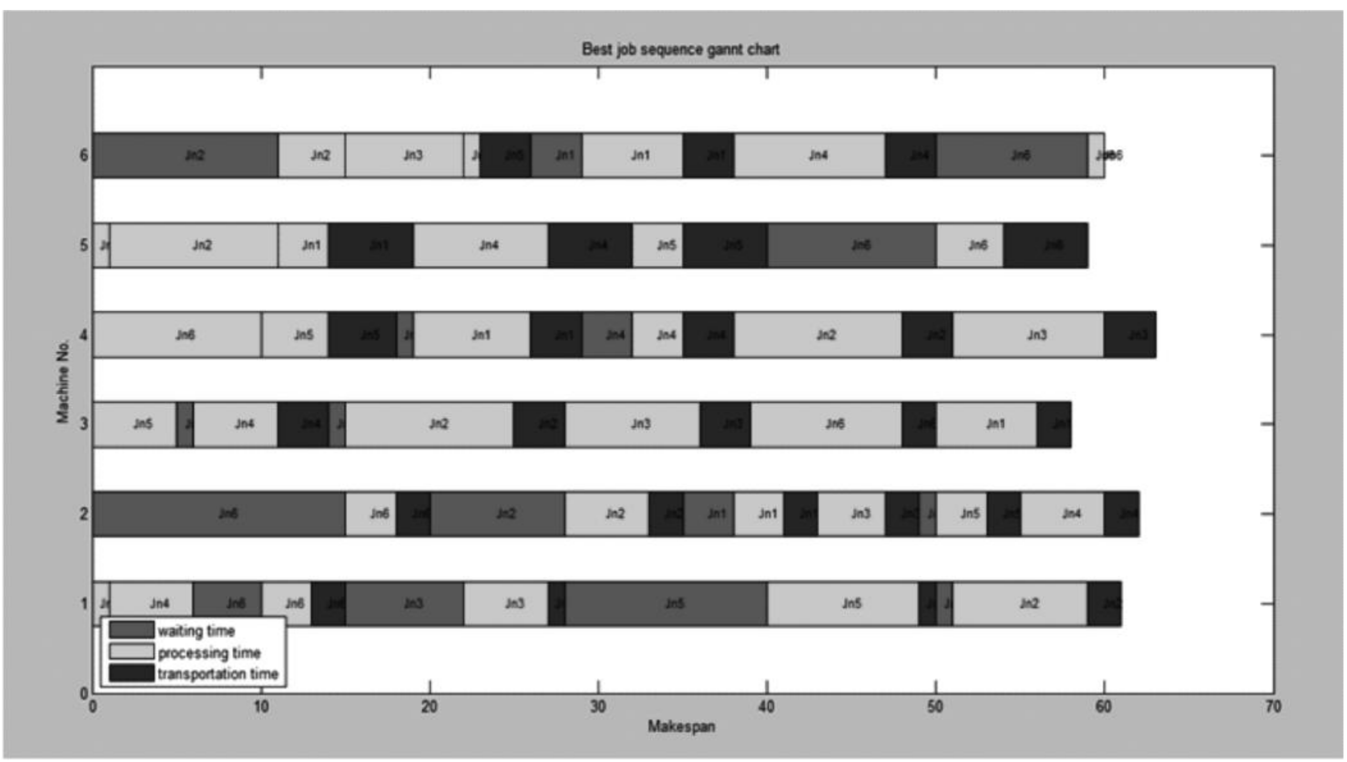

Fig. 4: Gantt chart

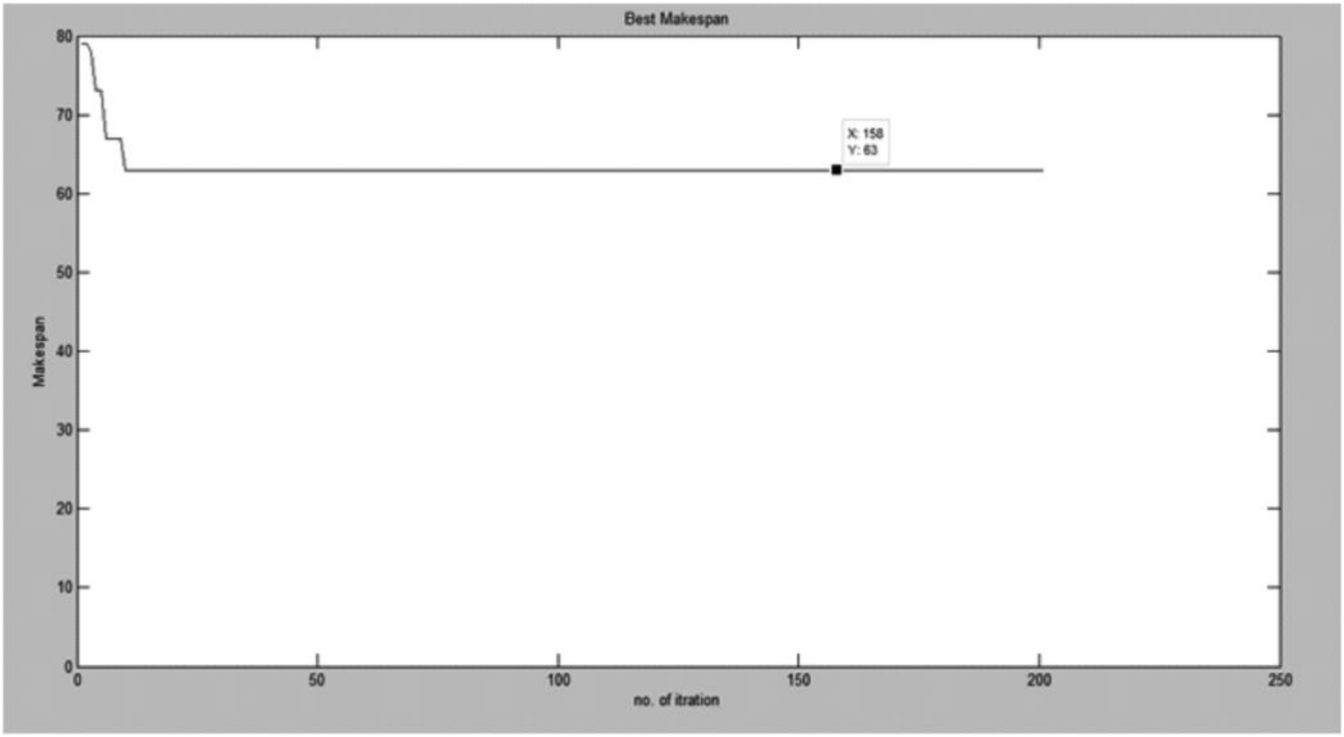

Fig. 5: Makespan vs Iteration

VI. CONCLUSION

This paper proposed a GA for solving the job shop scheduling to minimize the makespan. The optimum objective value obtained by GA for case study is represented by Figure.2 and makespan of given problem decreases from 55 units to 47 units with 15% improvement of the makespan. The addition of transportation time in shortest processing time increases the value of makespan. Further work includes considering other meta-heuristics (Branch & Bound, Integer linear Programming, Taboo Search and Simulated Annealing) for the job shop scheduling problem.

REFERENCES

[1] Ren Qing-dao-er-ji and Wang Yuping (2013). A new hybrid genetic algorithm for job shop scheduling problem. Computers & Operations Research 39:2291–2299.

[2] Sadrzadeh Amir (2013). To Solving the job shop scheduling problem via a modified PSO. Technical Journal of Engineering and Applied Sciences TJEAS Journal-2013-3-1/59-64.

[3] Sierra María R., Mencía Carlos, and Mencía Ramiro Mencía (2013). A new schedule generation schemes for the job-shop problem with operators. J Intell ManufDOI 10.1007/s10845-013-0810-6.

[4] Zhang Rui, Song Shiji, and Wu Cheng (2013). "A dispatching rule-based hybrid genetic algorithm focusing on non-delay schedules for the job shop scheduling problem". Int J Adv Manuf Technol 67:5–17

[5] Li J., Pan Q., Xie S., and Jun-qing S. Wang (2012).A Hybrid Artificial Bee Colony Algorithm for Flexible Job Shop Scheduling Problems.Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844 Vol. VI, No. 2 (June), pp. 286-296

[6] Teekeng Wannaporn, and Thammano Arit (2012).Modified Genetic Algorithm for Flexible Job-Shop Scheduling Problems. Procedia Computer Science 12:122 – 128.

[7] Zhang Guohui, Gao Liang, and Shi Yang (2011). An effective genetic algorithm for the flexible job-shop scheduling problem. Expert Systems with Applications 38. 3563–3573.

[8] 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.

[9] Defersha Fantahun M. and Chen Mingyuan (2010). A parallel genetic algorithm for a flexible job-shop scheduling problem with sequence dependent setups. Int J Adv Manuf Technol 49:263–279.

[10] Lei Deming (2010). Solving fuzzy job shop scheduling problems using random key genetic algorithm. Int J Adv Manuf Technol 49:253–262.

[11] Sun Liang, Cheng, Xiaochun, and Liang Yanchun (2010). To Solving for Job Shop Scheduling Problem Using Genetic Algorithm with Penalty Function, International Journal of Intelligent Information Processing Volume 1, Number 2.

[12] Zhang Rui, Wu Cheng (2010). A hybrid approach (HA) to large-scale job shop scheduling. Appl Intell 32: 47–59.

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

[14] Jinwei Gu, Xingsheng Gu (2009). A novel parallel quantum genetic algorithm for stochastic job shop scheduling. Journal of Mathematical Analysis and Applications J. Math. Anal. Appl. 355 63–81.

[15] Rossi Andrea and Boschi Elena (2009). A hybrid heuristic (HH) method to solve the parallel machines job-shop scheduling problem. Advances in Engineering Software 40:118–127.

[16] Chaoyong Zhang & Yunqing Rao (2008). An effective hybrid genetic algorithm for the job shop scheduling problem. Int J Adv Manuf Technol 39:965–974.

[17] Yong Ming Wang (2008). A two-stage genetic algorithm for large size job shop scheduling problems. Int J Adv Manuf Technol. 39:813–820.

[18] Jie Gao, Mitsuo Gen and Linyan Sun (2006). Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. J Intell Manuf 17:493–507.

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

[20] Omar, M., Baharum, A., & Hasan, Y. A. (2006). A job-shop scheduling problem (jssp) using genetic algorithm (ga). Regional conference on Mathematics, Statics and Applications, University Sains Malasiya, Penang, 13-15 June 2006.

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

[22] 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.

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

[24] Pinedo, M., (2001) Scheduling: Theory, Algorithms, and Systems, Prentice Hall, New York.

[25] Amico Delland Trubian Marco (1993). Annals of Operations Research - A Special issue on Taboo Search. Volume 41 Isse 1-4, Pages 231-252.

[26] Garey, M. R., & Johnson, D. S. (1979). Computer and intractability: A guide to the theory of NP-completeness. San Francisco: W.H. Freeman.

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

❐