

# IJERT

ISSN : 2278-0181

## International Journal of Engineering Research & Technology

Publish & Find Papers @



[www.ijert.org](http://www.ijert.org)

 **BROWSE**

OPEN  ACCESS

Call for Papers

# GENETIC ALGORITHM FOR THE FLOWSHOP SCHEDULING PROBLEM

<sup>1</sup>Vinit Saluja, <sup>2</sup>Rajeev Choudhary

<sup>1,2</sup> Assistant Professor, Geeta Engineering College, Panipat

<sup>1</sup>saluja\_vinit123@rediffmail.com

<sup>2</sup>rajeevmegec@gmail.com

**Abstract:** In this paper, we have introduced the concepts of a Genetic Algorithm and we described a Genetic Algorithm –based heuristic for solving the flowshop scheduling problems. The flowshop scheduling problem is a production problem where a set of  $n$  jobs have to be processed with identical flow patterns on  $m$  machines. Heuristics play a major role in solving NP –hard combinatorial optimization problems. This paper describes a Genetic Algorithm – based heuristic to makespan minimization on flowshop scheduling. Computational results indicate that optimal makespan can be achieved from various operation sequences of job instead of one.

## I. INTRODUCTION

Scheduling is the process of generating the schedule and schedule is a physical document which generally tells the happening of things and shows a plan for timing of certain activity. Scheduling can also be defined as the process of assigning a set of tasks to resources over a period of time (Pinedo, M., 2001). Effective scheduling plays a very important role in today's competitive manufacturing world. Performance criteria such as machine utilization, manufacturing lead times, inventory costs, meeting due dates, customer satisfaction, and quality of products are all dependent on how efficiently the jobs are scheduled in the system. Both exact and approximate methods are available for scheduling but exact methods of scheduling tend to fail due to enormous computational work with the increase in number of jobs and machine. Thus approximate methods are generally used to solve scheduling problems. A flow shop is characterized by a unidirectional flow of work, i.e. all jobs have the same processing order through all the machines. In simple flow shop, each machine centre includes just one machine and in flexible flow shop for at least one machine centre or stage, there exists more than one machine available for processing. The flow shop problem consists of 'm' machines and 'n' jobs and the scheduler objective is to find an optimal ordering of 'n' jobs (Baker et al., 2009). Flow shop scheduling has been widely used in many industrial applications such as manufacturing of variety of printers in computer industry, tyre manufacturing industry and printed circuit board manufacturing industry etc.

We study the flow-shop problems considering the following assumptions:

- All jobs are available for processing at time zero.

- Machines never breakdown and available throughout the scheduling period.
- No machine may process more than one operation at the same time.
- Infinite buffer exist between stages and before the first and after the last stage.
- Job processing cannot be interrupted.
- A job cannot be processed on more than one machine at the same time.
- No pre-emption is allowed.
- Travel time is not taken into consideration.
- A job follows precedence constraint of the operation

Several heuristic approaches for the flowshop scheduling problem have been developed. In recent years, meta-heuristic approaches, such as Simulated Annealing, Tabu Search, and Genetic Algorithms, have become very desirable in solving combinatorial optimization problems because of their computational performance.

## II. LITERATURE REVIEW

Several researchers have addressed the problem of flow shop scheduling. Some important contributions are discussed below.

Riad Aggoune et al (2001) proposed a genetic algorithm for flow shop scheduling problem with availability constraint to minimize the makespan and the total weighted tardiness problems. Most of the papers on scheduling take the common assumption that the machines are always available. In this paper, they consider a flow shop problem with availability constraints (FSPAC), in which unavailability times of the machines are known in advance as a preventive maintenance activity. Marimuthu et al (2005) evaluated heuristic search algorithm with lot streaming for a two machine flow shop problem with multiple jobs to minimize makespan. Three heuristic search algorithm are evaluated i.e. Bakers Algorithm (BA), Genetic Algorithm (GA) and Simulated Annealing (SA) algorithm. Result indicates that the GA and SA outperform the Bakers Algorithm. Among GA and SA, GA outperforms SA in almost all problems. Marimuthu et al (2008) proposed two evolutionary algorithms namely genetic algorithm (GA) and hybrid evolutionary algorithm (HEA) for scheduling  $n$  jobs  $m$  machine flow shops under lot sizing environment. The objective is to minimise the makespan/flow time. The performance of proposed GA

and HEA are compared with Baker's algorithm (BA) and Simulated Annealing (SA) algorithm. Computational result shows that the proposed algorithm GA and HEA are capable of producing optimal solution. Yaurima Victor et al (2008) proposed a modified Genetic Algorithm to minimise makespan of hybrid flow shop with unrelated machines, sequence dependent time and availability constraint. Authors found that the algorithm gives 6.27% better value of the objective function in average. Jarboui Bassem et al. (2011) proposed a hybrid genetic algorithm for solving no wait flow shop scheduling problem to minimise makespan and the total flow time. The variable neighborhood search (VNS) is used as an improvement procedure in the last step of the genetic algorithm (GA). VNS helps to prevent the algorithm from being stuck into local optima. Pang King Wah (2012) presented a Genetic Algorithm based heuristic approach to solve the problem of two machine no wait flow shop scheduling with setup time on machine is class dependent and the objective is to minimize the maximum lateness of the job processed. This class of machine scheduling problem has many practical applications in manufacturing industry such as metal refinery operation, food processing industry, chemical products and production processes in which no interruption between subsequent processes is allowed and the product can be grouped into families. Wang Li Chih et al. (2013) studied a solar cell industry scheduling problem, which is similar to traditional hybrid flowshop scheduling (HFS). However, the challenge in solar cell manufacturing is the number of machines that can be adjusted dynamically to complete the job. An optimal production scheduling model is developed to explore these issues, considering the practical characteristics, such as hybrid flowshop, parallel machine system, dedicated machines and sequence dependent job setup times. The objective of this model is to minimise the makespan.

### III. GENETIC ALGORITHM

Genetic Algorithms were developed by Holland in 1975. The Genetic Algorithm (GA) is a search technique based on the mechanics of natural Genetics and survival of the fittest. GA simulates the biological processes that allow the consecutive generations in a population to adapt to their environment. In the Simple GA-based approach, the various stages like evaluation, selection, crossover and mutation are repeatedly executed after initialization until a stopping criterion is met. The algorithm works on multiple solutions simultaneously. The Genetic Algorithm object determines which individuals should survive, which should reproduce, and which should die. Since Genetic Algorithms (GAs) are adaptive and flexible, the GAs were shown to be successfully applied to several optimization problems. For example, they have been applied to routing, scheduling, adaptive control, game playing, cognitive modelling, transportation problem, travelling salesman problems,

optimal control problems, database query optimization, etc.

### General Methodology

The general procedures of the GA are as follows:

1. Initialize a population of binary or non-binary chromosomes.
2. Evaluate each chromosome in the population using the fitness function.
3. Select chromosome to mate (reproduction).
4. Apply Genetic operators (crossover and mutation) on chromosome selected.
5. Put chromosomes produced in a temporary population.
6. If the temporary population is full, then go to step 7. Otherwise; go to step 3.
7. Replace the current population with the temporary population.
8. If termination criterion is satisfied, then quit with the best chromosome as the solution for the problem. Otherwise, go to step 2.

### IV. Adopted Methodology

In order to find the optimal schedule for a flow shop with sequence dependent setup times, a genetic algorithm based methodology is adopted in the present work. This section represents the details of the adopted methodology.

#### a) Representation

Encoding is the first step of GA. Each feasible solution is encoded as a chromosome (string or individual) also called a genotype (encoded solution). Various encoding schemes are available for flow shop scheduling as discussed in previous chapter. The selected encoding scheme should be able to represent the desired information. The present work utilizes permutation encoding for representation. In this method, strings (chromosome) are coded as a sequence of numbers (genes) with each gene representing one of the operations of job involved. The specific operation represented by the genes is interpreted according to the order of the gene in the chromosome. The length of chromosome depends upon the number of jobs/parttypes and the number of stages available in flow shop on which jobs are to be processed. Figure 1 represents the chromosome structure for a production order consisting of 4 parttypes to be processed in flowshop having 2 stages.

1 <sup>st</sup> gn	2 <sup>nd</sup> gn	3 <sup>rd</sup> gn	4 <sup>th</sup> gn	5 <sup>th</sup> gn	6 <sup>th</sup> gn	7 <sup>th</sup> gn	8 <sup>th</sup> gn
1	3	1	4	2	3	2	4
1 <sup>st</sup> op	1 <sup>st</sup> op	2 <sup>nd</sup> op	1 <sup>st</sup> op	1 <sup>st</sup> op	2 <sup>nd</sup> op	2 <sup>nd</sup> op	2 <sup>nd</sup> op

Legend: gn – gene, op – operation

Figure 1: Structure of chromosome

*b) Initialization*

In this step, initial population is generated having a fixed number of chromosomes and it is called population size. Initial population contains suitable number of solutions for the problem. Generally, initial population is generated randomly (Deb, K., 2006). The present work considers a population size equal to 10 and it is generated randomly.

*c) Evaluation of fitness function*

Each chromosome gives a measure of fitness via a fitness function (evaluation or objective). It is performance evaluation of chromosomes. GA is naturally suitable for solving maximization problems (Deb, K., 2006). Since objective functions in the present work is minimization of makespan  $[f(x)]$ , this minimization problem is transformed into maximization problem by following transformation rule (Deb, K., 2006)

$$F(x) = 1 / [1+f(x)]$$

where  $f(x)$  = value of makespan,

$F(x)$  = fitness function

*d) Selection*

Selection operator determines which chromosomes undergo for crossover and mutation. This decision is based on fitness of the chromosomes. During selection, the fitness of chromosomes is compared in order to choose the better chromosomes to drive search in good region of search space. Various selection methods such as Roulette Wheel Selection, Tournament Selection and Rank Selection can be used for selection in Genetic Algorithm. In the present study, Tournament selection is used.

*e) Crossover*

Crossover is used as the main Genetic operator and the performance of a GA is heavily dependent on it. A crossover operator is used to recombine two strings to get a better string. It is important to note that no new strings are formed in the reproduction phase. In the crossover operator, new strings are created by exchanging information among strings of the mating pool. A crossover operator is mainly responsible for the search of new strings even though mutation operator is also used for this purpose sparingly. In this study, two point crossover with a crossover probability of 0.85 is used. Due to above crossover methodology, some illegal off springs may generate in the above method. Then repairing is done to resolve the illegitimacy of off spring after mutation.

*f) Mutation*

Mutation is nearly always regarded as an integral part of a GA. Mutation generates an offspring solution by randomly modifying the parent's feature. It helps to preserve a reasonable level of population diversity, and

provides a mechanism to escape from local optima. In the present work, swap mutation with mutation probability of 0.15 is used.

*g) Repairing*

As discussed earlier, some illegal off springs may generate during crossover. For this, repairing is needed to resolve the illegitimacy of off springs after mutation. A repairing procedure is utilized for this purpose. It checks the string from left to right. If at any point, some genes repeats more than required and some genes are missing, then excess genes at any place are replaced by missing genes randomly.

*h) Elitism*

After generating offspring's, the parent strings of previous generation may get completely replaced. The best individuals can be lost in two cases (i) if, they are not selected to reproduce and/or (ii) they are destroyed by crossover or mutation (Mitchell, M., 2002). Thus, elitism strategy is used in order to force GA to retain some number of the best individuals at each generation. In the present work, elitism transfers a good individual from previous population to population of next generation with the elitism rate of 0.9 and it means that 10% best population is carried on into the next generation. For example, if, population size is 10, then the total number of best individuals from previous generation to be carried into next generation is equal to one.

*i) Termination Criterion*

It refers to the stopping criterion for further exploration in the search space. In the present work, if the fitness value does not change for 150 iterations, GA terminates and algorithm reaches to the optimum value of makespan

## V. Experimental Setup

In the present work, an attempt is made to optimize the Flow Shop Scheduling problem with sequence dependent setup time using Genetic Algorithm approach. The following parameters are used to calibrate genetic algorithm based heuristic: Population size: 10; Population generation: Random; Crossover type: Two point crossover; Crossover probability: 0.85; Mutation type: Swap mutation; Mutation probability: 0.15. The numbers of jobs are 12 and processing time is drawn from the uniform distribution U [5-75]. Quantity of each job to be manufactured is also drawn from uniform distribution U [15-30]. Setup time is sequence dependent and drawn from the uniform distribution U [5-55]. The number of stages is set to be 5 and each stage has 2 machines which are identical. The condition for the termination is 150 iterations without improvement of objective function. The above methodology is coded in MATLAB<sup>®</sup> and executed on Windows platform on Intel (R) Core 2 Duo CPU T6400@ 2.00 GHz 2.00 GHz, 4 GB RAM.

Five simulation runs are taken for the present problem and optimization yield best makespan as well as the operation sequence of jobs. Fig 2 shows the convergence curve for the present problem

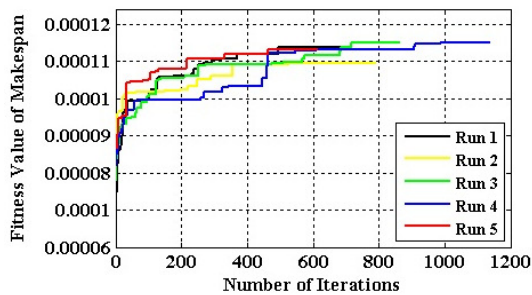


Figure 2: Convergence plot

## VI. CONCLUSIONS AND FUTURE SCOPE:

In the present work, an attempt is made to solve flow shop scheduling problem with sequence dependent setup times using genetic algorithm approach. From the analysis of the problem considered we have concluded that optimal makespan can be achieved by various operational sequences instead of one.

The present work can be extended in several ways. The above problem can be solved using other meta-heuristic techniques such as Tabu search, Simulated annealing, Fuzzy logic and comparison can be made for efficiency of genetic algorithm with these techniques. The problem can also be extended by incorporating the aspect of transportation time. The problem can also be extended by varying population size, crossover probability, mutation probability, selection pressure and the result can be compared. Moreover, the problem can also be extended by taking more than one objective simultaneously i.e. multi objective optimization

## VII. REFERENCES

- [1]. Allahverdi, Ali; Ng, C.T.; Cheng, T.C.E.; & Kovalyov Y. Mikhail, "A survey of scheduling problems with setup times or costs", *European journal of Operational Research*, 2008, Vol. 187, page. 985-1032.
- [2]. Abyaneh, Sina Hakimzadeh; and Zandieh, M., "Bi objective hybrid flow shop scheduling with sequence dependent setup times and limited buffers", *International Journal of Advanced Manufacturing Technology*. 2012, Vol. 58, page 309-325.
- [3]. Baker, Kenneth; and Trietsch, (2009), "Principal of Sequencing and Scheduling", John Wiley & Sons.
- [4]. Deb, K., (2003), "optimization for Engineering design, Algorithm and Examples", Prentice Hall, New Delhi, India.
- [5]. Goldberg, D.E., (1989), "Genetic Algorithm in search, optimization & machine learning", Pearson Education.
- [6]. Graves, S. "A review of production scheduling" *Journal of Operation Research*, 1981, Vol. 29, page 646-675.
- [7]. "Minimizing makespan in a flow shop with batch processing machines using simulated annealing", *Journal of Robotics and Computer Integrated Manufacturing*, Vol. 25, page 667-679.
- [8]. Marimuthu, S. and Ponnambalam, S.G., "Heuristic search algorithms for lot streaming in a two machine flowshop", *International Journal of Advanced Manufacturing Technology*. 2005, Vol. 27, page 174-180.
- [9]. Marimuthu, S.; Ponnambalam, S.G.; and Jawahar, N. "Threshold accepting and ant-colony optimization for scheduling m-machine flow shops with lot streaming", *Journal of Materials Processing technology*, 2009, Vol. 209, page 1026-1041.
- [10]. Marimuthu, S.; Ponnambalam, S.G.; and Jawahar, N. "Evolutionary algorithms for scheduling m machine flow shop with lot streaming" *Journal of Robotics and Computer Integrated Manufacturing*, 2008, Vol. 24, page 125-139.
- [11]. Mitchell, M., (2002), "An introduction to genetic algorithm", Prentice Hall, New Delhi, India.
- [12]. Obitko, Marek, (1998), "Introduction to genetic algorithm", [www.obitko.com/tutorials/geneticalgorithm/](http://www.obitko.com/tutorials/geneticalgorithm/).
- [13]. Pang, King wah. "A genetic algorithm based heuristic for two machine no wait flow shop scheduling problems with class setup times that minimize maximum lateness ", *International Journal of production Economics*, available online, 2012, <http://dx.doi.org/10.1016/j.ijpe.2012.06.017>.
- [14]. Pan, Q. K., "A discrete artificial bee colony algorithm for the lot streaming flow shop scheduling problem", *Journal of Information sciences*, 2011, Vol. 181, page 2455-2468.
- [15]. Pinedo, M. (2001), "Scheduling: Theory, Algorithm, and Systems", Prentice Hall, New York.
- [16]. Pinedo, M. L. (2005), "Planning and Scheduling in manufacturing and services", Springer, New York.
- [17]. Ruiz, Ruben; and Maroto, Concepcion, "A genetic algorithm for hybrid flowshops with sequence dependent setup times and machine eligibility", *European Journal of Operational Research*, 2006, Vol. 169, page. 781-800.
- [18]. Ventura, Jose A; and Yoon Suk Hun, "A new genetic algorithm for lot streaming flow shop scheduling with limited capacity buffers", *Journal of Intelligent Manufacturing*, 2012, DOI 10.1007/s10845-012-0650-9
- [19]. Wang, Shijin; and Liu, Ming, "A genetic algorithm for two stage no wait hybrid flow shop scheduling problem", *Journal of Computers & Operations Research*, 2013, Vol.40, page 1064-1075.
- [20]. Xiao, W.; Hao, P.; Zhang, S. & Xu, X., "Hybrid flow shop scheduling using genetic algorithm", 2000, *Proceeding of the 3rd world congress on Intelligent Control and Automation*, China
- [21]. Yaurima Victor; Burtseva Larisa; and Tchernykh Andrei, "Hybrid flowshop with unrelated machines, sequence dependent setup time and availability constraint: an enhanced crossover operator for a genetic algorithm" *Production planning and advanced manufacturing*, 2008, page 608-617.