Improved Solution to Job Shop Scheduling Problem with Delay Constraints using Genetic Algorithm

Swati Singh
Computer Science Dept
IMS Engineering College

Shruti Kapoor
Computer Science Dept
IMS Engineering College

Shelly Chikara
Computer Science Dept
IMS Engineering College

Vijai Singh
Astt Professor, CS
IMS Engineering College

ABSTRACT
Job shop scheduling problem is one of the most difficult NP-hard combinatorial optimization problems. Therefore, determining an optimal schedule and controlling and it is considered a difficult task. To achieve high performance in manufacturing firms, a scheduling system should make the right decision at the right time according to system conditions. It is difficult for traditional optimization techniques to provide the best solution. This paper focuses on the problems of determination of a schedule with the objective of minimizing the total make span time. An attempt has been made to generate a schedule using Genetic Algorithm.

Keywords
Scheduling, Genetic Algorithm, Makespan, Machine utilization

1. INTRODUCTION
To find the best schedule can be very easy or very difficult, depending on the shop environment, the process constraints and the performance indicators [1]. The Job-shop Scheduling Problem (JSP) is considered to be one the most complex combinatorial problems in computer science belonging to the class of NP Complete problems. In JSP, a set of jobs must be processed on a set of machines. Where each job is formed by a sequence of successive operations, every operation requires exactly one machine for a specified time, and machines are constantly obtainable and can process one operation at a time without pause. The complexity of the problem statement advocates for obtaining an optimal schedule in reasonable time by the use of heuristic techniques, instead of looking for an exact solution. Various approaches have been proposed for scheduling such as branch and bound, priority rules, Tabu search,
simulated annealing, genetic algorithms etc. In recent years, the adoption of meta-heuristics like GA has led to better results than classical dispatching or heuristic algorithms [2]. Solving scheduling problems with GA methods have been introduced by many researchers.
The rest of the paper is organized as follows. Section 2 presents the literature review pertaining to jssp. In section 3, the concept of Genetic algorithm is presented. In Section 4, we have defined our problem using Genetic Algorithm method to find the optimal schedule and its functioning. Finally, conclusions and the scope for further improvement are given in the last section

2. LITERATURE SURVEY
A comparative study of the latest algorithms and techniques has been done to gain a better insight into the problem solution space. The comparison was carried out while keeping in mind various parameters such as, selection, mutation, crossover, fitness criteria and solution space. After in- depth analysis we came to a conclusion that there is a tradeoff between optimality of solutions and computation time required. Since JSSP is a NP complete problem, there exists no specific algorithm which would give the optimal solution for all cases. We observed that, whenever a constrained JSSP was taken, the solutions obtained were near optimal when compared to the results obtained by modified crossover techniques [3].

2.1 Comparison of Algorithms:

	Selection	**Crossover**	**Mutation**	**Fitness criteria**	**Infeasible solution**	**Characteristic**
The penalty method[4] for constraints in JSSP	Random	Normal	Hyper-mutation	High threshold value to replace k worst chromosomes by k best	Taken	Constrained JSSP using Penalty function
Generalized order crossover [5]	Neighborhood mating and local offspring acceptance	GOX i.e. permutation with repetition	Position based mutation	Active scheduling	Not taken	Generalized order crossover
The job-based order crossover (JOX) [6]	Random	JOX	Shift change	Ranking method	Taken(transformation from infeasible to active)	Static JSSP and Job based order crossover
Hybrid genetic algorithm using random keys [7]	Random and top 10% direct	Parameterized uniform crossover	Replacing bottom 20% by random generation	Minimum makespan function	Not taken	Priority of operation and delay time
Variable neighborhood descent algorithm for flexible job shop scheduling problems [8]	Ranking based	Order-based	Allele based and immigration based	Roulette wheel selection	Not taken	VND, local search space optimization, flexible JSSP
Order-based Griffer and Thompson algorithm [9]	50%ND+ 50%GT	Order-based	Sublist permutation using order-based	Fitness value taken	Not taken	Static JSSP

3. GENETIC ALGORITHM

In the field of Computer Science "Genetic Algorithm" is a heuristic search technique that mimics the process of natural evolution to generate useful solutions to combinatorial and optimization problems. Genetic Algorithm belongs to a set of algorithms called Evolutionary Algorithms which solve a problem by taking a population of solutions and applying genetic operators in each reproduction. GA proposed by John Holland [9], uses the basic Darwinian mechanism of "survival of the fittest" and repeatedly utilizes the information contained in the solution population to generate new solutions with better performance. The presence or absence of genes and their order in the chromosome decide the characteristics of a species. Different traits are passed on from one generation to the next through different biological processes that operate on the genetic structure. He also demonstrated that a computer simulation of this process could be employed for solving optimization problems. In a GA, each solution is stored in an artificial chromosome represented by a string of binary bits or numbers. New candidates are generated gradually from a set of renewed populations by applying artificial genetic operators selected from policies based on the survival of the fittest principle, after repeatedly using operators of crossover and mutation [10].

A typical GA process consist of following steps

Step 1: Generate the initial population. Determine the size of the population and the maximum number of the generation.

Step 2: Calculate the fitness value of each member of the initial population.

Step 3: Calculate the selection probability of each member of the initial population using the ratio of fitness value of that initial population to the summation of the fitness values of the individual solutions.

Step 4: Select a pair of members (parents) that can be used for reproduction using selection probability.

Step 5: Apply the genetic operators such as crossover, mutation, and inversion to the parents. Replace the parents with the new offspring to form a new population.

Check the size of the new population. If it is equal to the initial population size, then go to step 6, otherwise go to step 4.

Step 6: If the current generation is equal to the maximum number of the generation then stop, else move to step 2.

4. PROBLEM DESCRIPTION

The problem is formulated considering 4 machines and 6 jobs with 2 numbers of pieces per job. The delay constraints involved during the manufacturing are also taken. The inputs are processing time and delay time. We have taken the processing time from a manufacturing unit in Northern India.

Details of input parameters used in our work are given in tables 1and 2

Table 1: Part processing time in minutes

	M1	**M2**	**M3**	**M4**
J1	40	50	60	70
J2	40	57	34	27
J3	60	54	80	36
J4	40	34	66	34
J5	49	23	34	20
J6	35	45	55	50

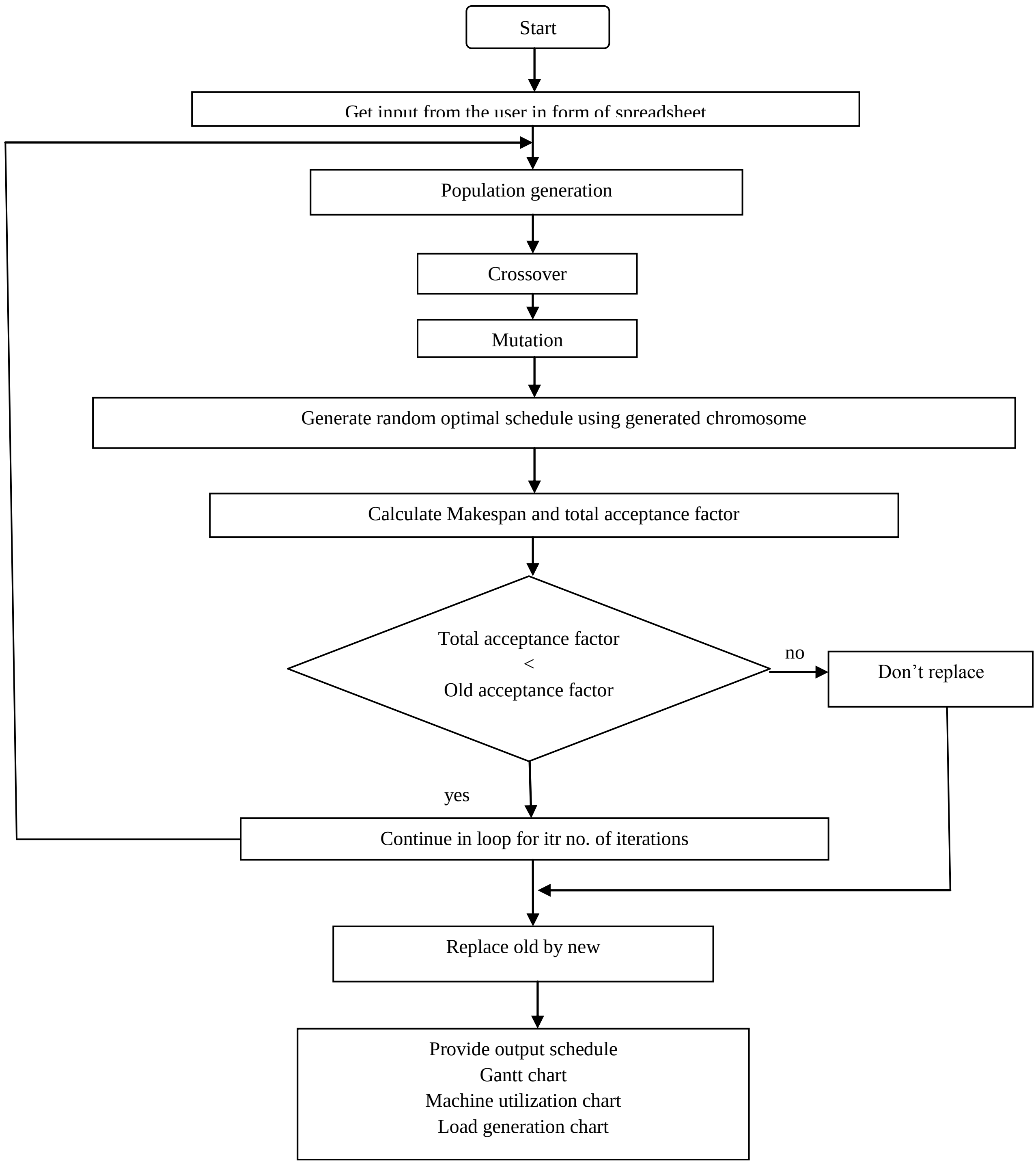
Table 2: Delay time in minutes in processing

	M1	**M2**	**M3**	**M4**
J1	10	12	8	7
J2	3	7	13	10
J3	6	4	10	12
J4	7	4	10	6
J5	1	4	8	10
J6	2	3	1	6

Following assumptions are made

1. The load/unload station capacity is unlimited.
2. Each machine completely manufactures the job assigned to it.
3. The jobs are atomic.
4. The inputs once set cannot be changed during the generation of the particular schedule.

The methodology can be summarized as follows:



4.1 Selection

The selection process selects chromosomes from the mating pool directed by the survival of the fittest concept of natural genetic systems. A chromosome is assigned a number of copies according to the selection process in which the chromosomes having minimum fitness value
(which is proportional to its fitness in the population) goes into the mating pool for further genetic operations. Random selection procedure is used.

4.2 Crossover

Single point crossover is used at the point ceil [p/2] *n, where p is the number of pieces per job and n is the total number of jobs.

4.3 Mutation

Each chromosome undergoes mutation with a fixed probability (P_m) that is called as mutation probability. Mutation maintains the diversity and improves the probability of selecting a chromosome from feasible solution set. Permutation mutation is used, in which the order of gene is changed. The transposition of gene depends upon the degree of mutation selected. For instance 100% mutation is achieved by complete permutation of the selected chromosome.

4.4 Parameter Setting

- Number of iterations 1000
- Crossover type: One point
- Mutation type: transposition
- Mutation probability 0.5

4.5 Algorithm- Chromosome Generation

```
function C = chromosome()
global  n p
C=zeros(1,n*p);
for i=0:p-1
   r=randperm(n);
   for j= 1:n
      C(j+i*n)=r(j);
   end
end
end
```

4.5 Algorithm- Crossover

```
function child= crossover(c1,c2)
for i=1:n*p
if i<= ceil(p/2)*
child[i]=c1[i];
else
child[i]=c2[i];
end
```

4.6 Algorithm- Mutation

```
function C= mutation(C)
global n p
z = randperm(n*p);
C = C(z);
end
```

4.7 Outputs obtained

Table3. JOB SEQUENCE ON EACH MACHINE

M1	J1	J2	J4
M2	J1	J5	J6
M3	J2	J5	J6
M4	J4	J3	J3

The above matrix displays each operation that would be done on a particular machine and the sequence of the same.

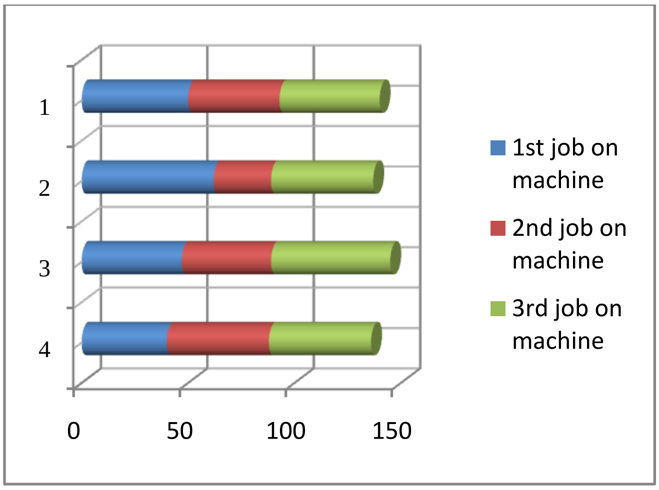


Fig 1 Gantt chart per machine

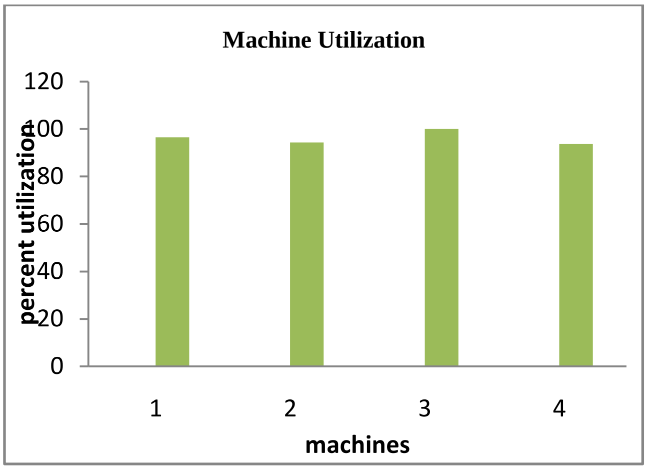


Fig 2 Machine utilization

It is the percentage utilization of each machine according to the generated schedule.

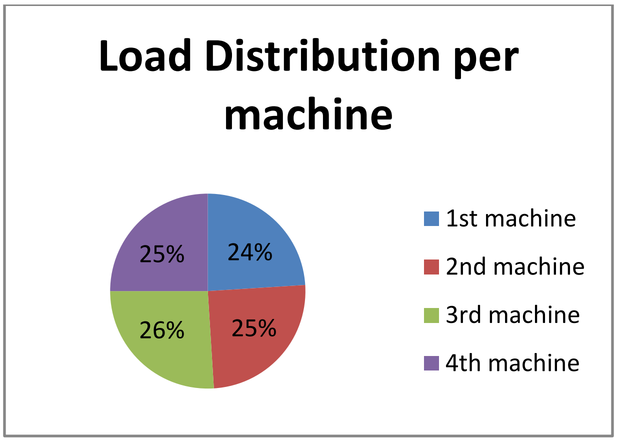


Fig 3. Load distribution per machine

Load distribution is the percentage of the total load that is on a particular machine.

5. CONCLUSIONS

The jssp many type of schedules depending upon the number of jobs and machines. Since the ability and functions of modern machines have been widely extended, the scheduling plan of a part might not be unique. So on a real shop floor, many feasible schedules can be found. It can be seen that Genetic Algorithm gives better result than traditional scheduling methods. The applicability of the GA based methodology has considerable potential application to manufacturing with further refinement in certain aspects, as outlined below.

1. We have considered transportation time and setup time as constant, further work can be done using setup time and transportation time.
2. We have assumed no break down, further work can be done taking into account actual breakdown.
3. The jobs are considered to be atomic. Further work can be done by creating schedules with various parts manufactured on different machines.

6. REFERENCES

[1] K. Deep et al. (Eds.): Proceedings of the International Conf. on SocProS 2011, AISC 131, pp. 875–882. springerlink.com © Springer India 2012

[2] Li, D., Chen, L., Lin, Y.: Using functional virtual population as assistance to learn scheduling knowledge in dynamic manufacturing environment. Internation Journal of Production Research 41(17), 4011-4024 (2003)

[3] Shruti Kapoor et al Ways and Means of Applying Genetic Algorithms for Job Shop Scheduling International Journal of Computational Intelligence Research ISSN 0973-1873 Volume 8, Number 1 (2012), pp. 19-25

[4] Liang Sun , Xiaochun Cheng, Yanchun Liang," Solving Job Shop Scheduling Problem Using Genetic Algorithm with Penalty Function" International Journal of Intelligent Information Processing Volume 1, Number 2, December 2010

[5] Christian Bierwirth in his paper "A Generalized Permutation Approach to Job Shop Scheduling Problem with Genetic Algorithms"

[6] Isao Ono Masayuki Yamamura Shigenobu Kobayashi in their paper, "A Genetic Algorithm for Job-shop Scheduling Problems Using Job-based Order Crossover" Proc. of ICEC'96, pp.547-552 (1996)

[7] José Fernando Gonçalves, Jorge José de Magalhães Mendes, Maurício G.C. Resende in their paper "A Hybrid Genetic Algorithm for the Job Shop Scheduling Problem"

[8] Jie Gao, Linyan Sun, Mitsuo Gen in their paper, "A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems" Computers & Operations Research 35 (2008) 2892 --- 2907

[9] J. H. Holland, Adaptation in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press, 1975.

[10] Goldberg, E. E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Reading, MA 1989