International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 12 (2015) pp. 32433-32450
© Research India Publications
http://www.ripublication.com

Production Scheduling Problem Solving With Sequence Dependent Set-Up Times By Using Genetic Algorithm

A.Muthiah [1] and R.Rajkumar [2]

[1] *Assistant Professor, Mechanical, PSR Engineering College, Sivakasi, Tamilnadu, India*

[2] *Professor, Mechanical, Mepco Schelnk Engineering College, Sivakasi, Tamilnadu, India*

Abstract

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. With redundant machines we have the security of knowing that we are not going to be in trouble meeting our deadlines if a machine has any unexpected down-times. Finally we can work to get our batch sizes as small as is reasonably possible while also reducing the setup time of each batch. This allows us to eliminate a sizable portion of each part waiting while the rest of the parts in the batch are being machined. We keep all of our machines well-maintained to prevent any problems, but there is on way to completely prevent down-time. Our research paper deals with the 5*5 job shop production scheduling problem solving with sequence dependant setup times considering the minimization of the maximum job completion time (Cmax) by using genetic operators with string evaluation method and genetic algorithm (GA) with Matlab simulation software method.

Index Terms: Job shop scheduling, minimization of maximum job completion time (Cmax), Genetic operators, Genetic algorithm (GA), Matlab simulation (Version R2009b).

Introduction

In any production facility where resources are shared between multiple batches or parts, it is extremely difficult to schedule all of the jobs so that no part is ever waiting at all. In fact this scenario is used to illustrate a difficult to impossible problem to solve with computers is known as the job shop problem. Scheduling in manufacturing systems is typically associated with allocating a set of jobs on a set of machines in order to achieve some objectives [1]. It is a decision making process that concerns the allocation of limited resources to a set of tasks for optimizing one or more objectives [2]. In manufacturing scheduling, Job-shop is a system that process n number of tasks on m number of machines. In this type of environment, products are made to order and in a low volume. Usually, these orders are differ in term of processing requirements, materials needed, processing time, processing sequence and setup times [3,4]. Job-shop problems are widely known as a NP-hard problem. Nowadays, search algorithms based on branch and bound methods and several approximation algorithms have been developed [5,6]. However, result from the branch and bound method sometimes is really unpredictable and requires a lot of time. It depends on the size of the problem [7]. Thus, schedulers are usually satisfied with an acceptance result which is not far from optima. Genetic algorithm is a search techniques that are widely used in industries [8,9]. A schedule is a suitable plan which generally tells the things that are made to happen; it shows us a plan for the timing of certain actions and answers the question, “When will a particular event take place” [10,11]. In language of industry, scheduling is a technique to order the jobs in a particular sequence. There are variety of sequencing rules which are followed in the industries such as first in first out basis, priority basis, job size basis and processing time basis etc [12,13].

The sequence is adapted which gives optimal or near optimal solution. Also scheduling is concerned with allocating limited resources to tasks to optimize certain objective functions [14]. Each task may have a certain priority level, an earliest possible starting time and a due date. In all the scheduling problems, the number of jobs and the number of machines are assumed to be finite [15]. The following approaches will be used to solve the job shop problem. With universal fixture system we can easily move the part to the most appropriate machine at the correct time and with minimal disruption.

Research Methodology

Planning and scheduling are distinctly different activities. The plan defines what must be done and restrictions on how to do it, the schedule specifies both how and when it will be done. The plan or schedule refers to the estimates of time and resource for each activity, as well as the precedence relationships between activities and other constraints. The schedule refers to the temporal assignments of tasks and activities required for actual execution of the plan. Job is a piece of work that goes through the series of operations. Shop is a place for manufacturing or repairing of goods or machinery. Scheduling is a decision process aiming to deduce the order of processing. Job shop scheduling problem means jobs to be processed on shop floor within specific time. JSSP is a very tough to solve. It is a very complex problem to solve. JSP is a working area where n jobs to be processed on m set of machines with many tasks. In JSP the machine order can be different for each job so it becomes complex and agile to get the optimal solution.

So genetic algorithm is used for our research study to attain the following objectives in job shop scheduling:

i. To minimize the makespan of the jobs, i.e., minimization of the maximum completion time and to minimize the processing cost.

ii. To minimize the maximal machine workload, i.e., the maximum working time spent at any machine. This objective is to prevent a solution from assigning too much work on a single machine and to keep the balance of work distribution over the machines.

iii. To minimize the total workload, which represents the total working time assigned over all machines. This objective is of interest if machine efficiency differ.

Genetic Algorithm

Genetic algorithms are evolutionary search techniques used to identify approximate solutions for optimization problems. They represents a computer simulation of a population of abstract representation called chromosomes of the candidate solutions called individuals to an optimization problem that evolves toward better solutions. The algorithm starts with a complete or partial randomly generated population. The evolution is simulated in generations. Each individual in this population has attached a fitness function that represents the individual performance based on a number of criteria. The new population is obtained from the old population by following three important steps: selecting the best individuals to become parents, performing crossover on the parents to obtain new individuals, and performing mutation to some very few individuals. The selection of individuals that will become parents is an important stage of the algorithm. Based on the fitness function value attached to each candidate, the individuals are chosen to become parents in order to increase the solution's quality.

Crossover and Mutation

Crossover operation is a genetic operator that aims to obtain the propagation of the best genetic material. The two chosen parents are combined and the resulted individuals are included in the new generation in order to increase the population performance. The mutation is represented by a chromosome modification applied to one or more genes. Figure 1 shows the schematic representation of genetic algorithm flow chart.

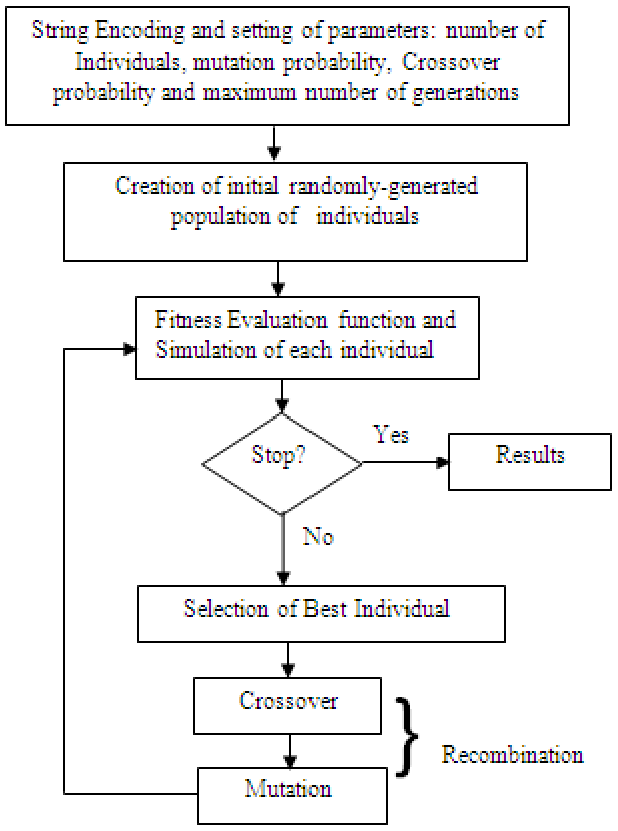
String Encoding and setting of parameters: number of Individuals, mutation probability, Crossover probability and maximum number of generations
Creation of initial randomly-generated population of individuals
Fitness Evaluation function and Simulation of each individual
Stop?
Yes
Results
No
Selection of Best Individual
Crossover
Mutation
Recombination

Fig.1. Genetic algorithm flow chart

Problem Description

The classical job-shop scheduling problem, J//Cmax is defined as follows: we are given a set of m machines {M1... Mm} and a set of n jobs {J1... Jn}.

Each job Jj, j=1,...,n, consists of a sequence of mj operations O1j,..., Omj,j, where Oij is an operation of job Jj to be processed on machine mij for a given uninterrupted processing time pij, where m(ij) i+ j≠1, for i=1,...,mj, j=1,...,n. The operations of each job must be processed in the given sequence.

Each machine Mi, i=1,...,m, can process at most one operation at a time, and at most one operation of each job Jj, j=1,...,n, can be processed at a time. Let Cij be the completion time of operation Oij. The objective is to get a schedule that minimizes the maximum completion time Cmax=max i,j Cij. A schedule is an allocation of a single time interval for each operation.

Encoding – Preference List Based Representation

How to encode solutions to chromosomes to ensure feasible solutions is a key issue for genetic algorithms. The preference list-based representation is used in our algorithm. In this encoding method the operations are arranged in a certain order. The sequence of operations of a job must stay intact also in the encoded solution. Table 1 shows a 5×5 instance; scheduled for achieve the smallest possible makespan.

Table 1 5*5 Job shop production scheduling input data's for problem solving

Jobs	Processing machines					Processing times				
	Process Orders					Process Orders				
	1	2	3	4	5	1	2	3	4	5
J1	M/C M1	M/C M2	M/C M3	M/C M4	M/C M5	32	78	56	40	15
J2	M/C M1	M/C M3	M/C M2	M/C M5	M/C M4	58	80	39	15	40
J3	M/C M2	M/C M1	M/C M5	M/C M3	M/C M4	90	85	75	30	11
J4	M/C M2	M/C M1	M/C M4	M/C M5	M/C M3	66	50	30	22	45
J5	M/C M1	M/C M3	M/C M2	M/C M5	M/C M4	76	49	57	28	25

From the Table 1, the operation sequence for each job
J1(J1 M1 32) (J1 M2 78) (J1 M3 56) (J1 M4 40) (J1 M5 15)
J2(J2 M1 58) (J2 M3 80) (J2 M2 39) (J2 M5 15) (J2 M4 40)
J3(J3 M2 90) (J3 M1 85) (J3 M5 75) (J3 M3 30) (J3 M4 11)
J4(J4 M2 66) (J4 M1 50) (J4 M4 30) (J4 M5 22) (J4 M3 45)
J5(J5 M1 76) (J5 M3 49) (J5 M2 57) (J5 M5 28) (J5 M4 25)

Job J1 must first be processed on machine M1 for 32 units. After that on machine M2 for 78 units and after that on machine M3 for 56 units and after that on machine M4 for 40 units and the last machine is M5 for 15 units. Similarly goes for the other four jobs. The schedule for this instance is encoded into a string, where the position of the operation in the string plays an important role. Operations are ordered with the help of a randomizer. Therefore it is important to use a reliable randomizer. String making in our case looks like this:

a) A list of first operations of all jobs is made.
((J1 M1 32) (J2 M1 58) (J3 M2 90) (J4 M2 66) (J5 M1 76))

b) One operation is chosen randomly from the list of first operations. That is (J5 M1 76). The operation is taken from the corresponding job and inserted into the string.
J1 (J1 M1 32) (J1 M2 78) (J1 M3 56) (J1 M4 40) (J1 M5 15)
J2 (J2 M1 58) (J2 M3 80) (J2 M2 39) (J2 M5 15) (J2 M4 40)
J3 (J3 M2 90) (J3 M1 85) (J3 M5 75) (J3 M3 30) (J3 M4 11)
J4 (J4 M2 66) (J4 M1 50) (J4 M4 30) (J4 M5 22) (J4 M3 45)
J5 (J5 M3 49) (J5 M2 57) (J5 M5 28) (J5 M4 25)
String:
((J5 M1 76))

c) Again a list of first operations of all the jobs is made. An operation is randomly selected from the list. That is (J3 M2 90). This operation is taken from the corresponding job and inserted as second operation in the string.
J1 (J1 M1 32) (J1 M2 78) (J1 M3 56) (J1 M4 40) (J1 M5 15)
J2 (J2 M1 58) (J2 M3 80) (J2 M2 39) (J2 M5 15) (J2 M4 40)
J3 (J3 M1 85) (J3 M5 75) (J3 M3 30) (J3 M4 11)
J4 (J4 M2 66) (J4 M1 50) (J4 M4 30) (J4 M5 22) (J4 M3 45)
J5 (J5 M3 49) (J5 M2 57) (J5 M5 28) (J5 M4 25)
String:
((J5 M1 76) (J3 M2 90))

d) The procedure is repeated until all the operations from the jobs are transferred into the string. If the procedure would be continued until the end, the string could look like this:
((J5 M1 76) (J3 M2 90) (J4 M2 66) (J1 M1 32) (J2 M1 58) (J2 M3 80) (J5 M3 49) (J1 M4 40) (J2 M4 40) (J5 M5 28) (J1 M5 40) (J4 M1 50) (J3 M1 85) (J1 M2 78) (J3 M5 75) (J2 M2 39) (J5 M2 57) (J1 M3 56) (J3 M3 30) (J4 M4 30) (J5 M4 25) (J4 M3 45) (J4 M5 22) (J3 M4 11) (J2 M5 15))

Job operations in the string still have the same processing order, but now they are mixed together. The string making is left to coincidence; it is possible to make a lot of versatile strings organisms which are necessary for the initial population.

String Evaluation

The feasibility is maintained throughout the searching process. Only feasible strings represent a solution to our problem. In our case the goal lies in the time optimization of schedules, we are interested in the makespan and processing order on the machines. For that reason, gantt charts are used for string evaluation and it is done step by step with adding operations one after another. The gantt chart and the makespan depend only upon the order in the string. In our case the operations are added directly from the string, from the left side to the right side. The operations which are at the beginning of the string have a higher processing priority than those at the end. This means, the gantt chart and the makespan depend only upon the order in the string. The operation order in the string is according to the precedence constrains. Otherwise the evaluation would give a false value. The gantt chart for our string is shown in fig.2. Numerical form of gantt chart is shown in Table 2. Table 3 shows the best possible schedule-1 sequence obtained from genetic operators with string evaluation method.

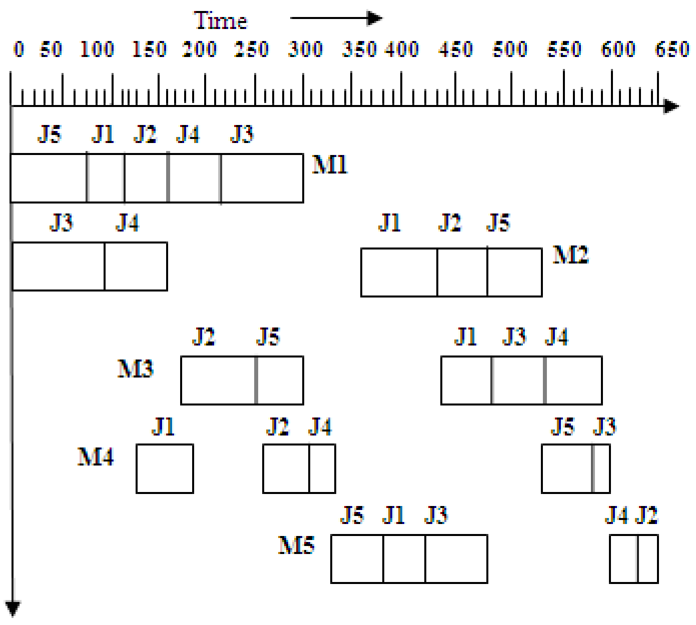

Fig.2. Best possible schedule-1 gantt chart for 5x5 instance string evaluation

Table 2 Machine output schedules in numerical form

Machines	Output schedule sequences in numerical form
M1	(0 J5 76) (76 J1 108) (108 J2 166) (166 J4 216) (216 J3 301)
M2	(0 J3 90) (90 J4 156) (363 J1 441) (441 J2 480) (480 J5 537)
M3	(166 J2 246) (246 J5 295) (441 J1 497) (497 J3 527) (527 J4 572)
M4	(108 J1 148) (246 J2 286) (286 J4 316) (537 J5 562) (562 J3 573)
M5	(295 J5 323) (323 J1 363) (363 J3 438) (572 J4 594) (594 J2 609)

Table 3 Job shop production scheduling (5*5) sequence results obtained from genetic operators with string evaluation method for jobs based on processing orders

Jobs	Processing order	Machine No.	Start Time	Finish Time
J1	1	M1	76	108
J1	2	M4	108	148
J1	3	M5	323	363
J1	4	M2	363	441
J1	5	M3	441	497
J2	1	M1	108	166
J2	2	M3	166	246
J2	3	M4	246	286
J2	4	M2	441	480
J2	5	M5	594	609

J3	1	M2	0	90
J3	2	M1	216	301
J3	3	M5	363	438
J3	4	M3	497	527
J3	5	M4	562	573
J4	1	M2	90	156
J4	2	M1	166	216
J4	3	M4	286	316
J4	4	M3	527	572
J4	5	M5	572	594
J5	1	M1	0	76
J5	2	M3	246	295
J5	3	M5	295	323
J5	4	M2	480	537
J5	5	M4	537	562

Genetic Operators

Genetic operators are used for solving a certain problem depends on the encoding method. The only genetic operation, which is independent from encoding, is selection. In our algorithm the tournament selection is used. Its purpose is to maintain the core of good solutions intact. This is done with transferring good solutions from one generation to the next one. The crossover operation often produces infeasible offspring, which are difficult to repair. The only genetic operation besides selection, which is used in our algorithm, is permutation. The permutation is based on switching operations inside the organism. The organism which will be permutated is chosen with the selection. The permutation procedure is described and shown on our string from Table 1:

a) A random operation is chosen from the string; let's say (J5 M1 76).
String:
((J5 M1 76) (J3 M2 90) (J4 M2 66) (J1 M1 32) (J2 M1 58) (J2 M3 80) (J5 M3 49) (J1 M4 40) (J2 M4 40) (J5 M5 28) (J1 M5 40) (J4 M1 50) (J3 M1 85) (J1 M2 78) (J3 M5 75) (J2 M2 39) (J5 M2 57) (J1 M3 56) (J3 M3 30) (J4 M4 30) (J5M4 25) (J4 M3 45) (J3 M4 11) (J4 M5 22) (J2 M5 15))

b) In our case, the right border is represented by the operation (J2 M3 85) and the left border is presented by the beginning of the string. In the string, the space between the borders is marked with square brackets.
String:
([(J5 M1 76) (J3 M2 90) (J4 M2 66) (J1 M1 32) (J2 M1 58)] (J2 M3 80) (J5 M3 49) (J1 M4 40) (J2 M4 40) (J5 M5 28) (J1 M5 40) (J4 M1 50) (J3 M1 85) (J1 M2 78) (J3 M5 75) (J2 M2 39) (J5 M2 57) (J1 M3 56) (J3 M3 30) (J4 M4 30) (J5 M4 25) (J4 M3 45) (J3 M4 11) (J4 M5 22) (J2 M5 15))

c) A random position between the brackets is chosen where the operation (J5 M1 76) is inserted: let us say in front of (J4 M2 66).
String:
((J3 M2 90) (J5 M1 76) (J4 M2 66) (J1 M1 32) (J2 M1 58) (J2 M3 80) (J5 M3 49) (J1 M4 40) (J2 M4 40) (J5 M5 28) (J1 M5 40) (J4 M1 50) (J3 M1 85) (J1 M2 78) (J3 M5 75) (J2 M2 39) (J5 M2 57) (J1 M3 56) (J3 M3 30) (J4 M4 30) (J5 M4 25) (J4 M3 45) (J3 M4 11) (J4 M5 22) (J2 M5 15))

This procedure randomly changes the chosen organism, which also changes the solution which the organism represents. Because there is often necessary to switch more than one operation in the organism, it is possible to repeat the whole procedure over and over. Table 4 presents the 5*5 output instance for possible schedule-1.

Table 4 Overall process sequencing output for 5*5 possible schedule-1 obtained from genetic operators with string evaluation method

Processing sequence for possible schedule-1	Job processing with respect to their machines and processing times
1,2,3,4,5	(J5 M1 76), (J3 M2 90), (J4 M2 66), (J1 M1 32), (J2 M1 58)
6,7,8,9,10	(J2 M3 80), (J5 M3 49), (J1 M4 40), (J2 M4 40), (J5 M5 28)
11,12,13,14,15	(J1 M5 40), (J4 M1 50), (J3 M1 85), (J1 M2 78), (J3 M5 75)
16,17,18,19,20	(J2 M2 39), (J5 M2 57), (J1 M3 56), (J3 M3 30), (J4 M4 30)
21,22,23,24,25	(J5 M4 25), (J4 M3 45), (J3 M4 11), (J4 M5 22), (J2 M5 15)

The 5*5 Job instance made up of 25 operations. The number of possible schedules S (solutions) can be calculated with Eq. (1).

$$S = (n!)^{m} \quad (1)$$

Where,
n = Total number of jobs
m = Total number of machines
For the 5*5 job instance case we get
$S = (5!)^5$
= 24883200000
S5*5 ≈ 24 million possible schedules.

GA with Matlab Simulation for Solving Job Shop Scheduling Problems

This section further focuses on solving a specific scheduling problem for job shop in industrial environment using genetic algorithm (GA) in Matlab simulation. The schedule is planned for job shop activities used in the industries. This Matlab simulation provided results in the form of job sequence, start and finish time for each activity considering the processing times.

Task Parameters of GA

i) Values of function domain must be transformed to the code strings.

ii) Do not directly transform task parameters, but their coded form.

iii) Lead searching, coming out not from one point, but from some population of points.

iv) Use only fitness function, but do not use derivative or other auxiliary information.

Design Principles of GA

Coding or Representation

String with all parameters

- Design alternative → individual (chromosome)
- Single design choice → gene
- Design objectives → fitness

Fitness function (Parent selection)

- Too strong fitness selection bias can lead to sub-optimal solution.
- Always keep the best one
- Too little fitness bias selection results in unfocused and meandering search

Reproduction Operators

Crossover

- Two parents produce two offspring.
- There is a chance that the chromosomes of the two parents are copied unmodified as offspring.
- There is a chance that the chromosomes of the two parents are randomly recombined (crossover) to form offspring.
- Generally the chance of crossover is between 0.6 and 1.0.
- Generating offspring from two selected parents
- Single point crossover
- Two point crossover (Multi point crossover)
- Uniform crossover

Mutation

- There is a chance that a gene of a child is changed randomly.
- Generally the chance of mutation is low.

Convergence (When to stop)

- After many generations, average fitness has converged, but no global maximum is found; not sufficient difference between best and average fitness.
- Relatively super-fit individuals dominate population
- Population converges to a local maximum.

Input and output parameters

Step 1: Input parameters

Step 1.1 No. of Jobs
Step 1.2 No. of Machines
Step 1.3 No. of Operations
Step 1.4 Machine Sequence for each jobs
Step 1.5 Processing Time for each operation of each job on each machine.
Step 1.6 Completion Time of each job
Step 1.7 Due time for each job as per priority
Step 2: Output Parameters
Step 2.1 Near optimal schedule (makespan value)
Step 2.2 Optimal job sequence
Step 2.3 Tardiness (Earliness and Lateness)

GA Parameters used while encoding Matlab

```
NIND=40; (Number of individuals)
MAXGEN=200; (Maximum number of generations)
GGAP=1; (Generation gap)
XOVR=0.8; (Crossover)
MUTR=0.009; (Mutation)
```

Processing Time

```
T= [32 78 56 40 15;
58 80 39 15 40 ;
90 85 75 30 11;
66 50 30 22 45;
76 49 57 28 25
];
```

Processing Machine

```
M=[ 1 2 3 4 5;
1 3 2 5 4;
2 1 5 3 4;
2 1 4 5 3;
1 3 2 5 4
];
```

GA with Matlab Simulation Results

Figure 3 shows the Gantt chart i.e.machine versus time chart for our 5*5 scheduling problem. In this chart processing times are plotted in X-axis and processing machines are plotted in Y-axis. By using this chart we can find out the overall production scheduling results with respect to each machines for each jobs. From the Gantt chart we obtain the output for 5*5 job shop scheduling. For the best possible schedule-2 (see Table 5) for machine wise schedule results based on priority level, (see Table 6) for job wise schedule results based on processed orders.

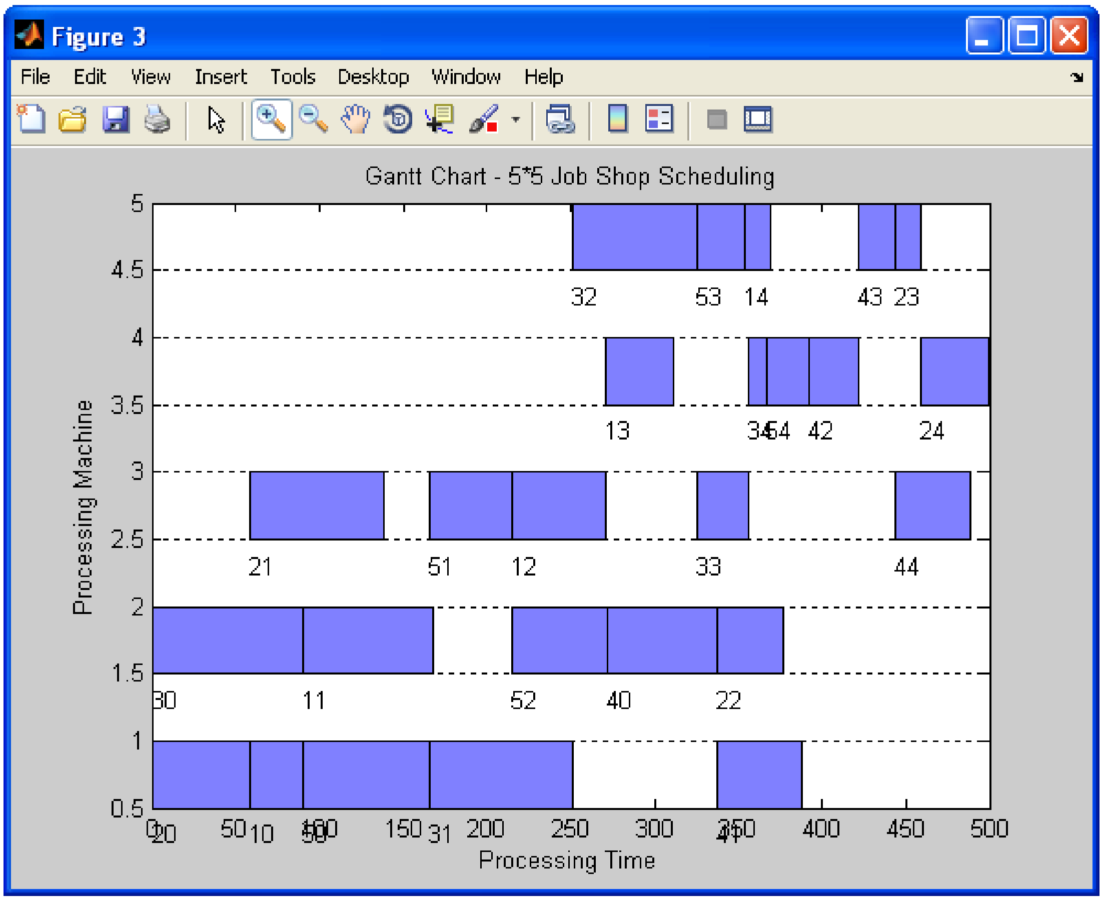

Fig.3. Matlab simulation best possible schedule-2 Gantt chart results

Table 5 Job shop production scheduling (5*5) sequence results obtained from Matlab simulation results for machines based on priority level

Machines	Process Priority	Job No.	Process No	Start Time	Finish Time
M1	1	J2	1	0	58
M1	2	J1	1	58	90
M1	3	J5	1	90	166
M1	4	J3	2	166	251
M1	5	J4	2	338	388
M2	1	J3	1	0	90
M2	2	J1	2	90	168
M2	3	J5	3	215	272
M2	4	J4	1	272	338
M2	5	J2	3	338	377
M3	1	J2	2	58	138
M3	2	J5	2	166	215
M3	3	J1	3	215	271

M3	4	J3	4	326	356
M3	5	J4	5	444	489
M4	1	J1	4	271	311
M4	2	J3	5	356	367
M4	3	J5	5	367	392
M4	4	J4	3	392	422
M4	5	J2	5	459	499
M5	1	J3	3	251	326
M5	2	J5	4	326	354
M5	3	J1	5	354	369
M5	4	J4	4	422	444
M5	5	J2	4	444	459

Table 6 Job shop production scheduling (5*5) sequence results obtained from Matlab simulation results for jobs based on processing orders

Jobs	Process No.	Machine No.	Start Time	Finish Time
J1	1	M1	58	90
J1	2	M2	90	168
J1	3	M3	215	271
J1	4	M4	271	311
J1	5	M5	354	369
J2	1	M1	0	58
J2	2	M3	0	58
J2	3	M2	338	377
J2	4	M5	444	459
J2	5	M4	459	499
J3	1	M2	0	90
J3	2	M1	166	251
J3	3	M5	251	326
J3	4	M3	326	356
J3	5	M4	356	367
J4	1	M2	272	338
J4	2	M1	338	388
J4	3	M4	392	422
J4	4	M5	422	444
J4	5	M3	444	489
J5	1	M1	90	166
J5	2	M3	166	215
J5	3	M2	215	272
J5	4	M5	326	354
J5	5	M4	367	392

Matlab simulation schedules are conducted for five different types of jobs on five different machines. Each types of the jobs are consists of five different tasks. So twenty five possible output processing sequences are listed in Table 7.

Table 7 Overall process sequencing output for 5*5 obtained from GA with Matlab simulation method

Processing sequence for possible schedule-2	Job processing with respect to their machines and processing times
1,2,3,4,5	(J2 M1 58), (J3 M2 90), (J1 M1 32), (J2 M3 80), (J5 M1 76)
6,7,8,9,10	(J1 M2 78), (J3 M1 85), (J5 M3 49), (J1 M3 56), (J5 M2 57)
11,12,13,14,15	(J3 M5 75), (J4 M2 66), (J1 M4 40), (J5 M5 28), (J3 M3 30)
16,17,18,19,20	(J4 M1 50), (J2 M2 39), (J3 M4 11), (J1 M5 15), (J5 M4 25)
21,22,23,24,25	(J4 M4 30), (J4 M5 22), (J2 M5 15), (J4 M3 45), (J2 M4 40)

Benchmarking Results

The problem is to find a suitable fitness function for a chromosome evaluation to get a solution for job shop scheduling problems. This suggests a new reasonable fitness function using GA techniques to evaluate population chromosomes efficiently. This technique used to give reward to the good chromosome and to apply a penalty on the bad chromosome. In order to approve the validity of the fitness function, another fitness function should be tested to get the results and compare the results with new fitness function results. Fitness Function that determine the fitness value according to the condition-action instances. Comparative analysis should be conducted and the comparison of various benchmark instance results are tabulated (see Table 8). Figure 4 shows the graphical representation of jobs priority level, maximum job completion time (Cmax) and maximum job waiting time for the possible schedules 1 and 2 comparison by using genetic operators with string evaluation method versus GA with Matlab simulation method respectively.

Table 8 Comparison of Benchmark Instances

Instances	Number of individuals	Max. Number of generations	Crossover Rate	Mutation rate	Total Possible Schedules
2*2	15	25	0.50	0.08	4
3*3	25	75	0.70	0.05	216
4*4	35	150	0.75	0.02	331776
5*5	40	200	0.80	0.009	24 million
6*6	55	250	0.85	0.008	1.39×10^{17}
7*7	70	300	0.90	0.007	8.26×10^{25}

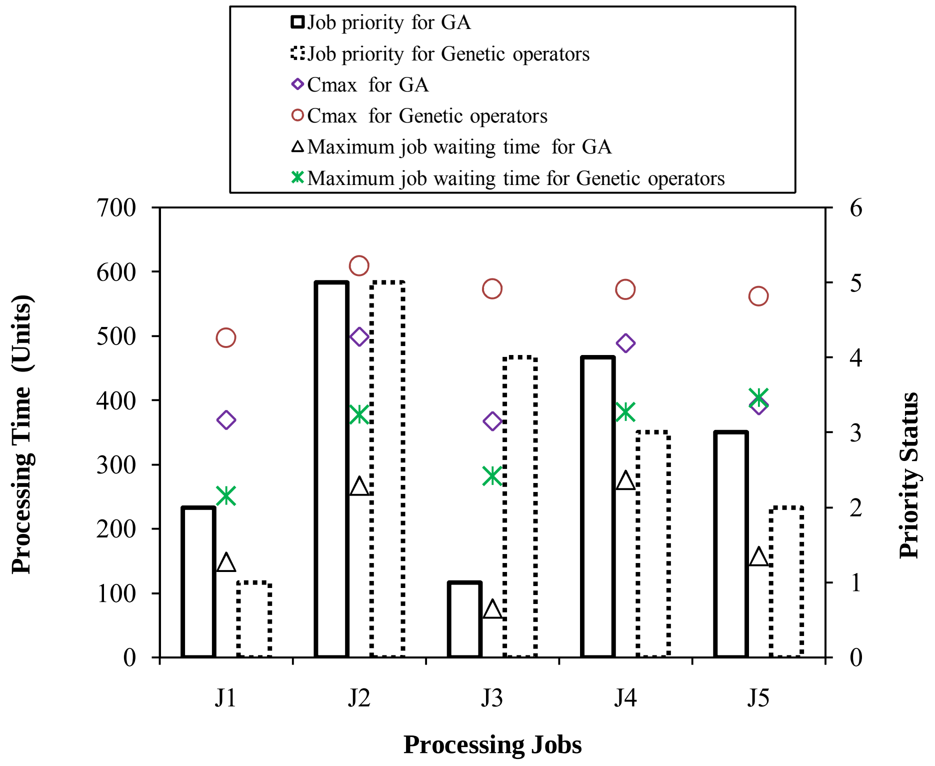

Fig.4. Graphical representation of genetic operators with string evaluation method versus GA in Matlab simulation method, (Cmax) and Maximum job waiting time

Conclusion

In this paper, we presented the genetic algorithm in Matlab R2009b simulation that are capable of finding good solutions for job shop production scheduling problem. Hence genetic operators with string evaluation method output schedule-1 is more machine idle time and maximum job completion time (Cmax) with an overall processing time of 0 to 609 units. Meanwhile GA with matlab simulation method output schedule-2 attain moderate machine idle time and maximum job completion time (Cmax) with an overall processing time of 0 to 499 units. Hence matlab simulation output effectively shows the smallest possible make span schedules to reduce the target of machine idle time and maximum job completion time (Cmax) effectively. So matlab output best possible schedule-2 is accepted due to shortest path for more efficiency. This algorithm would be an upgrade to a hybrid algorithm. Quality of final solution depends on the initial population and pure chance. So that search procedure has to be repeated several times for a specific problem.

So genetic algorithm is proved for matlab simulation to attain the following aspects effectively in job shop production industries by considering the following aspects:

i) *Make span of the job minimization:* Maximum job completion time is minimized (Cmax) and processing cost is also minimized.

ii) *Minimization of maximal machine work load:* Maximum working time spent at any machine is minimized and equal work load is distributed for all machines.

iii) *Minimization of total machine work load:* Total processing time for all work assigned machines is reduced so total machine idle time is decreased.

iv) *GA with Matlab simulation:* It is also applicable for all recommended complicated job shop scheduling problems for all types of industries.

Future Scope

Also, research papers are classified based on various criteria of GA such as its parameter selection, computer resource usage, hybridization and enhancement from the past work. These days further work is being continuously performed in developing new hybrid GA algorithms by using various methods such as combination of GA or other optimizing methods thus developing better results than these results are again rectified counter checked and taken best results out of many using methods like Design of Experiment (DoE) and many more, which finally identify future research scope in this widely growing area.

References

[1] Jamili, A., Shafia, M.A., Tavakkoli-Moghaddam, R. (2011), “A hybridization of simulated annealing and electromagnetism-like mechanism for a periodic job shop scheduling problem”, Expert Systems with Applications, 38 (5), pp 5895-5901.

[2] Jia, Z.Y., Ma, J.W., Wang, F.J., Liu W. (2011), “Hybrid of simulated annealing and SVM for hydraulic valve characteristics prediction”, Expert Systems with Applications, 38(7), pp 8030–8036.

[3] Lin, S.W., Ying, K.C., Lu, C.C., Gupta, J.N.D. (2011), “Applying multi-start simulated annealing to schedule a flow line manufacturing cell with sequence dependent family setup times”, International Journal of Production Economics, 130(2), pp 246-254..

[4] Wang, J.B., Li, J.X. (2011), “Single machine past-sequence-dependent setup times scheduling with general position-dependent and time-dependent learning effects’, Applied Mathematical Modelling, 35, pp 1388–1395.

[5] Yu, W., Liu, Z., Wang, L., Fan, T. (2011), “Routing open shop and flow shop scheduling problems”, European Journal of Operational Research, 213(1), pp 24-36.

[6] Choi, B.C., Leung, J.Y.T., Pinedo, M.L. (2010), “A note on makespan minimization in proportionate flow shops”, Information Processing Letters, 111(2), pp 77–81.

[7] Czapinski, M. (2010), “Parallel simulated annealing with genetic enhancement for flow shop problem with Csum”, Computers and Industrial Engineering, 59(4), pp 778-785.

[8] Eren, T. (2010), 'A bi-criteria M-machine flow shop scheduling with sequence-dependent setup times", Applied Mathematical Modelling, 34(2), pp 284-293.
[9] Li, S., Ng, C.T., Cheng, T.C.E., Yuan, J. (2010), "Parallel-batch scheduling of deteriorating jobs with release dates to minimize the makespan', European Journal of Operational Research, 210(3), pp 482–488.
[10] Mason, S.J., Chen, J.S. (2010), "Scheduling multiple orders per job in a single machine to minimize total completion time". European Journal of Operational Research, 207(1), pp 70–77.
[11] Safari, E., Sadjadi, S.J. (2010), "A hybrid method for flow shops scheduling with condition-based maintenance constraint and machines breakdown", Expert Systems with Applications, 38(3), pp 2020–2029.
[12] Wang, J.B., Wang, M.Z. (2010), "Single machine multiple common due dates scheduling with learning effects", Computers and Mathematics with Applications, 60(11), pp 2998–3002.
[13] Zhao, C., Tang, H. (2010), "Single machine scheduling with past-sequence-dependent setup times and deteriorating jobs', Computers & Industrial Engineering, 59(4), pp 663–666.
[14] Chiang, T.C., Fu, L.C. (2009), "Using a family of critical ratio-based approaches to minimize the number of tardy jobs in the job shop with sequence dependent setup times", European Journal of Operational Research, 196(1), pp 78–92.
[15] Gholami, M., Zandieh, M., Alem-Tabriz, A. (2009), "Scheduling hybrid flow shop with sequence- dependent setup times and machines with random breakdowns", International Journal of Advance Manufacturing Technology, 42(1-2), pp 189-201.
[16] Naderi, B., Zandieh, M., Roshanaei V. (2009), "Scheduling hybrid flow shops with sequence dependent setup times to minimize makespan and maximum tardiness', International Journal of Advance Manufacturing Technology, 41(11-12), pp 1186–1198.
[17] Rad, S.F., Ruiz, R., Boroojerdian, N. (2009), "New high performing heuristics for minimizing makespan in permutation flow shops', OMEGA, 37(2), pp 331-345.
[18] Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov, M.Y. (2008), "A survey of scheduling problem with setup times or costs", European Journal of Operational Research, 187(3), pp 985-1032.
[19] Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K. (2008), "A simulated annealing-based multi objective optimization algorithm: AMOSA", IEEE Transactions on Evolutionary Computation, 12(3), pp 269-283.
[20] Eren, T. (2007), "A multi criteria flow shop scheduling problem with setup times", Journal of Materials Processing Technology, 186(1-3), pp 60–65.24.
[21] Luo, X., Chu, C. (2007), "A branch-and-bound algorithm of the single machine schedule with sequence-dependent setup times for minimizing maximum tardiness", European Journal of Operational Research, 180(1), pp 68–81.

[22] Vinod, V., Sridharan, R., 2011. Simulation modeling and analysis of due-date assignment methods and scheduling decision rules in a dynamic job shop production system. International Journal of Production Economics 129,127–146.

[23] Al-Hinai, N., ElMekkawy, T., 2011. An efficient hybridized genetic algorithm architecture for the flexible job-shop scheduling problem. Flexible Services and Manufacturing Journal 23, 64–85. doi:10.1007/s10696-010-9067-y.

[24] Chaari T, Chaabane S, Loukil T, Loukil D (2011). A genetic algorithm for robust hybrid flow shop scheduling. Int. Jou. Comput Integr Manuf 24:821–833.

[25] Ye LI and Yan CHEN from Transportation Management College, Dalian Maritime University, Dalian, PR.China in Journal of Software Vol.5, No.3 March2010.A Genetic Algorithm for Job Shop Scheduling.

[26] A. Allahverdi, H. Aydilek, Heuristics for the two-machine flow shop scheduling problem to minimize maximum lateness with bounded processing times, Comput. Math. Appl. 60 (2010) 1374–1384.