

## Optimizing Makespan In JSSP Using Unordered Subsequence Exchange Crossover In GA

Sureshkumar.S<sup>1</sup>, Saravanan.G<sup>2</sup>, Dr. S. Thiruvankadam<sup>3</sup>

<sup>1</sup>Department of Computer science and Engineering, P.A College of Engineering and Technology, Pollachi

<sup>2</sup>Assistant Professor, P.A College of Engineering and Technology, Pollachi

<sup>3</sup>Department of information technology Head/Professor, P.A College of Engineering and Technology, Pollachi

---

**Abstract:** The objective of this paper is to minimize the makespan time in job shop scheduling problem. The JSSP is a one of the optimization problem in computer science and production environment. In order to minimize the makespan time and find out the optimal schedule special crossover technique is used i.e. Unordered Subsequence Exchange Crossover (USXX) in Genetic Algorithm (GA). Using the special cross over technique USXX the most of the benchmark results are compared and obtain the results near to optimal value of the benchmark problems.

**Keywords-**JSSP; Makespan time; Genetic Algorithm; USXX;

---

### I. INTRODUCTION

We introduce the job shop problem is a well known problem in the fields of operational research, industrial engineering and computers science. This problem is one of the important optimization problems because it is used in the most planning and managing of manufacturing processes. One of the most common scheduling problems is job shop scheduling problem (JSSP), where a set of independent jobs must be processed on a set of available machines. Each job is a sequence of operations. Each operation requires a predefined machine. There are two different jsp environments they are static environment in which planning horizon is predeterminedly assigned before the schedule has to be taken. In contrast of static scheduling is called dynamic scheduling in which machines and jobs are assigned while implementing the schedule. for example in case of machine is breakdown that time the schedule is stopped to resolve this dynamic environment is supported by changing the machines were the given schedule is processing. This paper concentrated based on the static environment that is the process routing is fixed to the given schedule. The routing is determined based on the permutation. Machine scheduling in the job shop case involves finding the sequence Shop based machine scheduling involves finding a sequence in which set of work orders (or) jobs is processed on each multiple machines. In this context, the order of production steps, also referred to as the **Process routing** of each job has to be considered. In job shop case, process routings are not required to be identical like in flow lines, hence they way be predefined individually for each job. In scientific literature, a production step is commonly referred to as an operation and the process routings are specified by so called **Precedence constraints**. The need for (properly) sequencing work orders on machines arises from the problem of resource contention that is when the workload induced by all currently available operations exceeds a machine's capacity at given time. The simplest form of introducing **Capacity constraints** is to allow only exclusive processing on machines, meaning that a machine can process at most one operation at a time.

#### 1.1 Over View of the Job Shop Scheduling Problem

Formally , a Job shop scheduling problem(JSSP) can be specified , by a finite set  $J$  of  $n$  jobs  $J=\{J_1, \dots, J_n\}$  which have to be scheduled on a finite set  $M$  of  $m$  Machines,  $M=\{M_1, \dots, M_m\}$ . Each job  $J_i$  is fragmented into a series of  $m$  operations  $O_{ik}$  , where subscript  $k$  indicates the machine  $M_k$  on which the operations has to be processed. The technological order of machines (process routing) for a job<sub>i</sub> is predefined; each operation  $O_{ik}$  is assigned a non-negative integer processing time  $p_{ik}$ . In general setting, an individual release time  $r_i$  may be associated with each job, yielding a so called dynamic job shop scheduling problem. However, to keep things simple, we stick to the static case, where all jobs are available from the beginning of planning horizon. Basically the scheduling goal is to determine a sequence of operations on each machine  $M_k$  such that precedence and capacity constraints of the problem are both satisfied. Although permutations of operations represent a solution to the problem, they are not practical with regard to post processing. Besides the position within the sequence, we are mainly interested in when a specific operation actually starts on a given machine. For this reason, a solution candidate is preferably described by a set  $S$  of operation starting times  $S_{ik}$ ,  $S = \{ S_{ik} | 1 \leq i \leq n, 1 \leq k \leq m \}$ .

**1.2 Types of Schedules**

A feasible schedule is called **semi active** if no operation can start earlier without changing the processing sequence on the respective machine. A feasible schedule is called **active** if no operation can be started earlier by changing the processing order on any of the machines without delaying some other operation. A feasible schedule is called **non-delay** if no machine is ever kept idle if there is at least one operation which is ready for processing.

**1.3 Example Problem**

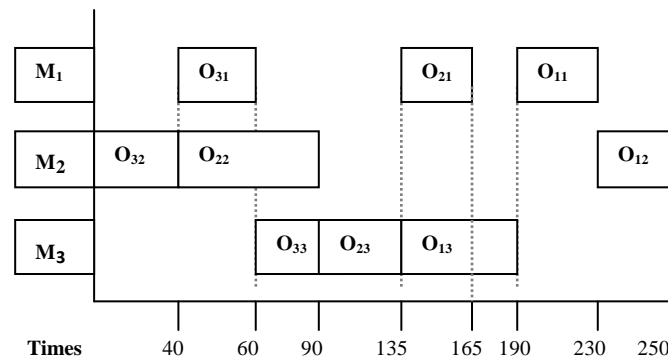
This problem contains 3 machines (m), 3 jobs (n), Process routing, processing time also given. If the operation sequence is given it will work based On the Precedence constraints and capacity constraints and satisfies the constraints of the problem also. Let we see how it works.

<b>Permutation Representation</b>	<b>Operation Sequence</b>
$J_1 = \{M_3, M_1, M_2\}$	$M_1 = (J_3, J_2, J_1)$
$J_2 = \{M_2, M_3, M_1\}$	$M_2 = (J_3, J_2, J_1)$
$J_3 = \{M_2, M_1, M_3\}$	$M_3 = (J_3, J_2, J_1)$

**TABLE I. OPEARTION PROCESSING TIMES**

JOBS	MACHINES		
	M <sub>1</sub>	M <sub>2</sub>	M <sub>3</sub>
J <sub>1</sub>	40	20	55
J <sub>2</sub>	30	50	45
J <sub>3</sub>	20	40	30

The above inputs the schedule information is showed in the Fig. 1. And makespan time is calculated using the Gantt chart.



**Fig. 1 Scheduled Visual information using Gantt chart**

For the above operation sequence or schedule and machine allocation or permutation, with operation time the makespan time is 250 calculated through Gantt chart. The scheduled information in Fig 1. Clearly showed satisfies the precedence and capacity constraints.

**II. Related Works**

Many Heuristics approaches are developed in the past decades by researches for solving job shop scheduling problems. Some of these methods are dispatch rules, tabu search, simulated annealing, particle swarm optimization. Genetic algorithm and artificial immune system. Christian Bierwirth proposed in order to sequence the tasks of a job shop problem on a number of machines related to technological order of jobs, a new representation technique mathematically known as Permutation with repetition is presented [4]. Khaled Mesghouni explains about how to use evolutionary algorithms to deal with a flexible job shop scheduling problem, especially minimizing the makespan [25]. Ramezanal Mahdavinejad provide solution to single processor job shop scheduling problems are solved by heuristic algorithm based on the hybrid of priority dispatching rules according to an ACO [24]. Kamrul Hasan et al. the JSSP is solved using GA. than priority rules are used to improve the performance of GA [23]. Moghaddas et al. (2008) presented an innovative mixed-integer linear programming using TSP and three different lower bounds are used [22]. Masaya yoshika et al. proposed a new genetic coding for the job shop scheduling problem [9]. Yan Chun Liang et al. present a GA with penalty function for the JSSP and two operations are used 1. Clonal selection based hyper mutation, 2. Lifespan extended strategy [19]. Allan Glaser and Meenal Sinha et al. Scheduling is important because much of our daily work requires us to juggle multiple activities in order to maximize productivity [2]. Yan Chen et al. analyze the dynamic JSSP when a machine breakdown and new job arrivals occur. And a university mathematical model for agile JSSP is constructed [10]. Surekha et al. solving JSSP GA and ACO is used with

Fuzzy parameters. In 2011 solving JSSP GA and PSO with Fuzzy processing time is used [14],[15]. Kanate ploydanai et al. developed a new algorithm based on non-delay scheduling heuristic by adding machine availability constraints to solve JSSP with minimize the makespan objective [6]. Mohammad Akhshabi et al. present a clonal selection algorithm to solve the flexible JSSP [11]. Thamilselvan et al. developed a different type of crossover approach to solve JSSP [18]. Mahmood Al Bashir et al. provide a rule base scheduler by deriving from an artificial neural network performing JSS [12]. Ahmed Tariq Sadiq et al. proposed an improved scatter search algorithm for solving JSSP and using three methods of algorithm to find out the makespan, finding minimum and solution combination method [1]. Mohammad et al. using the artificial immune algorithm with clonal selection principle [21]. Babukarthik et al. proposed to minimize the makespan using the ACO and Cuckoo search for JSSP [3]. Nikumbh et al and Nedi karimi Na sab et al. solve a JSSP using simulated annealing (SA) [13].

### III. PROPOSED ALGORITHM

The objective of proposed scheme is to solve a job shop scheduling problem to minimize the makespan time. In order to solve a JSSP artificial intelligence technique genetic algorithm (GA) is used. The genetic algorithm is a probabilistic Meta heuristic technique, which is used to solve optimization problems. They are based on the genetic process of chromosome. Over many generations, natural population evolves according to the principles of natural selection that is survival of the fittest. It starts with the initial solution called population and it is filled with chromosome. Each element in chromosome is called gene. Job is represented by each gene in chromosome and the job sequence in a schedule based on the position of the gene. In our proposed algorithm unordered subsequence exchange crossover (USXX) and shift change Mutation is used.

#### 3.1 Constraints of the proposed Algorithm to solve JSSP

1. Each job visits each machine exactly once.
2. The operations of a job have to be processed strictly sequential in the given order.
3. No overlapping or parallel processing is allowed.
4. Machine setup times and transportation times are neglected.
5. Each operation is assigned exactly one single machine for processing.
6. Machines are assumed to be continuously available.

#### 3.2 Objective function

The main objective of the JSSP is to find a schedule of operations the can minimize the maximum completion time called makespan, that is the completed time of carrying total operations out in the schedule for n jobs and m machines. The objective or fitness function takes the input as the number of jobs, number of operations, chromosome, operation time sequence and machine sequence of the corresponding operation. "Eq. (1)" The fitness function produces the output as a makespan value for the corresponding operation sequence.

$$C_{\max} = \max_{1 \leq i \leq n} C_i \rightarrow \min \quad (1)$$

#### 3.3 Chromosome representation and decoding

Chromosomes are initialized by n\*m real numbers. For instance 4 jobs and 3 machines mean the chromosome as follows. (4\*3=12 chromosome required for this 4\*3 JSSP).

Chromosome representation

10.2	2.5	7.5	6.7	1.3	5.3	2.7	7.4	7.3	4.1	4.5	10.5
------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------

Ranking

11	2	10	7	1	6	3	9	8	4	5	12
----	---	----	---	---	---	---	---	---	---	---	----

Decoding by using the formula,

$$(R_i \bmod n) + 1 \quad (2)$$

Where,

$R_i$  is integer number by ranking.

n, is number of jobs.

Using "Eq. (2)" from the ranking obtain the results shows,

4	3	3	4	2	3	4	2	1	1	2	1
---	---	---	---	---	---	---	---	---	---	---	---

Operation sequence is find out based on the above values that is 4 job 1 operation ( $O_{41}$ ), 3 job 1 operation ( $O_{31}$ ), 3 job 2 operation ( $O_{32}$ ) likewise it is goes up to 12<sup>th</sup> operation.

From the chromosome we obtained the operation sequence.

$O_{41}$	$O_{31}$	$O_{32}$	$O_{42}$	$O_{21}$	$O_{33}$	$O_{43}$	$O_{22}$	$O_{11}$	$O_{12}$	$O_{23}$	$O_{13}$
----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------	----------

### 3.4 Algorithm steps

- Step1 : initialize the number of chromosomes by generating n\*m real numbers for each chromosome.
- Step2 : Assign the operation time sequence and Machine sequence for selected chromosomes.
- Step3 : find the makespan value for each and every chromosome using the objective function and also find the minimum makespan value among different values.
- Step4 : select N/2 chromosomes using the Roulette- Wheel selection from different chromosomes.
- Step5 : Applying the Unordered subsequence exchange crossover and shift change Mutation to generate the new chromosomes.
- Step5 : find the makespan values for newly generated chromosomes using the objective function.
- Step7 : choose the best chromosomes which have the minimum makespan values from newly generated and also from old chromosomes.
- Step8 : find the minimum makespan value among different chromosomes.
- Step9 : terminates if the maximum number of iteration is reached or optimal value is obtained.

### 3.5 Unordered subsequence exchange crossover (USXX)

Unordered subsequence exchange crossover creates a new children's even the subsequence of parent 1 is not in the same order in parent 2. the algorithm for USXX is as follows.

- Step1 : select the two parents from the different chromosomes initialized with the sequence of all operation. Say p1 and p2.

<b>P1</b>	4	3	3	4	2	3	4	2	1	1	2	1
<b>P2</b>	1	4	3	2	3	4	2	1	3	4	1	2

- Step2: Generate two children from P1 and P2 name it as C1 and C2.
- Step3: select the gene from P1 and same gene selected in P2 but it should unordered position in P2

<b>P1</b>	4	3	3	4	2	3	4	2	1	1	2	1
<b>P2</b>	1	4	3	2	3	4	2	1	3	4	1	2

- Step4: crossover started from P1 that is P2 unselected genes are move to the p1 in unselected position. So C1 is generated. Likewise to generate C2 crossover from P2 to P1.

<b>C1</b>	4	3	3	1	4	2	3	2	1	4	1	2
-----------	---	---	---	---	---	---	---	---	---	---	---	---

### 3.6 Shift change Mutation

Shift change mutation is implemented by shift the same job index in every place into the same direction so that new sequence is generated.

4	3	3	1	4	2	3	2	1	4	1	2
---	---	---	---	---	---	---	---	---	---	---	---

After applying the Mutation, the new sequence is obtained, then evaluate the fitness function for this sequence.

4	3	3	1	2	4	2	3	1	4	2	1
---	---	---	---	---	---	---	---	---	---	---	---

## IV. Experimentatl Results

The proposed algorithm is implemented using java programming language on windows platform with Intel Pentium dual core. Table II summarizes the results such as problem name, problem size, optimal solution and makespan.

TABLE II. MAKESPAN VALUES FOR DIFFERENT BENCHMARKS

<i>Si no</i>	<i>Problem name</i>	<i>Problem size</i>	<i>Optimal solution</i>	<i>Proposed algorithm makespan</i>
1	FT06	6 X 6	55	55
2	LA02	10 X 5	666	666
3	LA01	10 X 5	655	739
4	LA10	15 X 5	958	958
5	LA11	20 X 5	1222	1222
6	LA13	20 x 5	1150	1160
7	LA14	20 X 5	1292	1292
8	LA15	20 X 5	1207	1207
9	LA16	10 X 10	945	1080
10	LA17	10 X 10	784	785
11	LA18	10 X 10	848	855
12	LA19	10 X 10	842	842
13	LA20	10 X 10	907	907

The results of the proposed algorithm are compared with the results of different benchmark problems. The optimal results of the benchmark problems are achieved in the proposed algorithm and some of the proposed results are found near to optimal. And used by the unordered subsequence exchange crossover the some of the benchmark problems results are reached within the minimum number of iterations.

### V. CONCLUSION AND FUTURE WORK

In this paper to solve the job shop scheduling problem objective is minimizing the makespan time using the unordered subsequence exchange crossover in genetic algorithm (GA). Each chromosome represents the feasible solution. Our approach is to generate an optimal schedule to minimize the makespan time. The proposed algorithm produced the chromosomes and operation sequence simple and easy compared to other algorithms, also proposed algorithm produces near optimal result for the benchmarks problems.

The proposed algorithm is based on the static environment to solve a job shop scheduling problem. Proposed algorithm results are compared with different benchmark problems, the results are obtained near to optimal only. Yet proposed algorithm does not guarantee to obtain an optimal result. In future the proposed algorithm can be combined with other soft computing techniques. This hybrid approach can provide a optimal result and can solve a job shop scheduling problem in dynamic environment with multi-criteria objective.

### ACKNOWLEDGEMENT

The authors are heartily thankful to their supervisor, Professor **Dr.S.Thiruvenkadam**, head of the department information technology, P.A. College of engineering and technology, whose sprit to work, guidance and support from the initial to the final level enabled them to develop an understanding of the subject. Above all and the most needed, he provided them unflinching encouragement and support in every possible ways.

The authors express their profound thanks and gratefulness to **Dr. D. Chitra**, head of the department, P.A College of engineering and technology for her advice and crucial contribution, which made the thesis possible to initiate.

Finally, the authors would like to thank everybody who were important and helped them out in every process to the successful realization of the thesis.

### References

- [1] Ahmed Tariq Sadiq and Kanar Shukr Mohammed, (March, 2012), "Improved Scatter Search for Job Shop Scheduling Problem", International Journal of Research and Reviews in Soft and Intelligent Computing, Vol.2, No. 1.
- [2] Allan Glaser and Meenal Sinha, (2010), "Scheduling Programming Activities and Johnson's Algorithm", NESUG.
- [3] Babukartik. R.G, Dhavachelvan. P, (August, 2012), "Hybrid Algorithm using the advantage of ACO and Cuckoo Search for Job Scheduling", International Journal of Information Technology Convergence and Services, Vol.2, No. 4.
- [4] Christian Bierwirth, (1995), "A generalized permutation approach to job shop scheduling with genetic algorithms", Volume 17,issue 2-3,pp 87-92.
- [5] Hiwa Farughi, Babak Yousefi Yegane, Hiresht Soltanpanah, Fayagh Zaheri, Foruzan Naseri, (2011), "Considering The Flexibility and Overlapping in Operation in Job Shop Scheduling Based on Meta-heuristic Algorithms", Australian Journal of Basic and Applied Sciences, 5(11): 526-533.
- [6] Kanate Ploydanai and Anan Mungwattana, (2010), "Algorithm for Solving Job Shop Scheduling Problem Based on machine availability constraint", International Journal on Computer Science and Engineering, Vol. 02, No. 05.
- [7] Liang Sun, Xiaochun Cheng, yanchum Liang,(December, 2010), "Solving Job Shop Scheduling Problem Using Genetic Algorithm with Penalty Function", International Journal of Intelligent Information Processing, Vol. 1, No. 2.
- [8] Mahdavinejad. R.A, (August, 2010), "A new approach to job shop-scheduling problem", Journal of Achievements in Materials and Manufacturing Engineering, Vol. 41, Issues 1-2.

- [9] Masaya Yoshikawa, Hideto Nishimura, Hidekazu Terai, (2009), "A New Genetic Coding for Job Shop Scheduling Problem Considering Geno Type", Recent Advances in Computer Engineering and Application.
- [10] Mehdi Karimi Nasab, Hamidreza Haddad and Payam Ghanbari, (2012), "A Simuated Annealing for the Single Machine Batch Scheduling with Deterioration and Precedence Constraints", Proc. Asian Journal of Industrial Engineering.
- [11] Mohammad Akhshabi, Mostafa Akhshabi and Javad Khalatbari, (2011), "Solving flexible job-shop scheduling problem using clonal selection algorithm", Indian Journal of Science and Technology, Vol.4, No.10.
- [12] Mohmood Al Badhir, Zahidul Islam. M.D, Masud. A.K.M, (May, 2011), "Approach to Job-Shop Scheduling Problem Using Rule Extraction Neural Network Model", Global Journal of Computer Science and Technology, Vol.11, Issue 7 Version 1.0.
- [13] Raich. A.R and Nikumbh. P.J, (2012), "Priority Based Flexible Job-Shop Scheduling Problem", BIOINFO Computer Engineering, Volume 2, issue 1, pp.22-24.
- [14] Surekha. P and Sumathi. S, (2011), "Solution to the Job Shop Scheduling Problem using Hybrid Genetic Swarm Optimization Based on  $(\lambda, 1)$ - Interval Fuzzy Processing Time", European Journal of Scientific Research, Vol. 64, No.2.
- [15] Surekha. P, Sumathi. S, (October, 2010), "Solving Fuzzy based Job Shop Scheduling Problems using Ga and Aco", Journal of Emerging Trends in Computing and Information Sciences, Vol 1, No. 2.
- [16] Tamilarasi. A and Anantha Kumar. T, (2010), "An enhanced genetic algorithm with simulated annealing for job-shop scheduling", International Journal of Engineering Science and Technology, Vol. 2, No. 1.
- [17] Thamilselvan Rakkiannan and Balasubramanie Palanisamy, (2012), "Hybridization of Genetic Algorithm with Parallel Implementation of Simulated Annealing for Job Shop Scheduling", Proc. American Journal of Applied Sciences. 9(10): 1694-1705.
- [18] Thamilselvan. R and Balasubramanie. P, (2011), "Analysis of Various Alternate Crossover Strategies for Genetic Algorithm to Solve Job Shop Scheduling Problems", European Journal of Scientific Research, Vol.64, No.4.
- [19] Ye LI and Yan CHEN, (March, 2010), "A Genetic Algorithm for Job-Shop Scheduling", Journal of Software, Vol. 5, No. 3.
- [20] Ye Li, Yan Chen, (September, 2010), "Hybrid Algorithm Approach To Job Shop Scheduling Problem", Global Joudullah rnal of Computer Science and Technology, Vol. 10, Issue 8 Ver. 1.0.
- [21] Zohreh Davarzani, Mohammad. R and Akbarzadeh. T, Nima Khairdoost, (July, 2012), "Multiobjective Artificial Immune Algorithm for Flexible Job Shop Scheduling Problem", International Journal of Hybrid Information Technology, Vol.5, No. 3.
- [22] Mohaddas. R, Houshmand. M, (March, 2008), "Job-Shop Scheduling Problem With Sequence Dependent Setup Times", Proceedings of the International Multi Conference of Engineers and Computer Scientists, Vol. II.
- [23] Kamrul Hasan. S.M, Ruhul Sarker and David Cornforth, (December, 2007), "GA with Priority Rules for Solving Job-Shop Scheduling Problems", New South Wales at the Australian Defence Force Academy.
- [24] Ramezanali Mahdavinejad, (2007), "A New Approach to Job Shop-Scheduling Problem", International Journal of Quality Research", Vol. 1, No. 1.
- [25] Khaled Mesghouni, Slim Hammadi, (2004), "Evolutionary Algorithms for Job-Shop Scheduling", International Journal Applied Mathematics Computing Science, Vol. 14, No.1.
- [26] Günther Zäpfel, Roland Braune, Michael Bögl, (2010), "Metaheuristic Search Concepts", ISBN 978-3-642-11342-0, Springer Heidelberg Dordrecht London New York.