International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958, Volume-9 Issue-1, October 2019

Effect of Crossover Probability on Performance of Genetic Algorithm in Scheduling of Parallel Machines for BI- Criteria Objectives

Raghavendra B V

Abstract*: Optimization of multi objective function gain the importance in the scheduling process. Many classical techniques are available to address the multi objective functions but the solutions yield the unsatisfactory results when the problem becomes complex and large. Evolutionary algorithm would be the solution for such problems. Genetic algorithm is adaptive heuristic search algorithms and optimization techniques that mimic the process of natural evolution. Genetic algorithms are a very effective way of obtaining a reasonable solution quickly to a complex problem. The genetic algorithm operators such as selection method, crossover method, crossover probability, mutation operators and stopping criteria have an effect on obtaining the reasonably good solution and the computational time. Partially mapped crossover operators are used to solve the problem of the traveling salesman, planning and scheduling of the machines, etc., which are having a wide range of solutions. This paper presents the effect of crossover probability on the performance of the genetic algorithm for the bi-criteria objective function to obtain the best solution in a reasonable time. The simulation on a designed genetic algorithm was conducted with a crossover probability of 0.4 to 0.95 (with a step of 0.05) and 0.97, found that results were converging for the crossover probability of 0.6 with the computational time of 3.41 seconds.

***Keywords:* Genetic Algorithm, Crossover probability, Bi-criteria objective, Scheduling, Parallel machines.**

I. INTRODUCTION

The Scheduling process is the crucial task in the ever changing production process to meet the customer demand at the earliest possible time. Scheduling is the process of allocation of resources over time and workload on the machines systematically to meet the objective function for optimization in the production process. The systematic allocation of workload and resources helps the decision makers in analyzing the effectiveness of the production process and deciding the best method. The scheduling is the complex task and helps to meet some of the objectives of the production process such as, to reduce manufacturing lead time, reduction in-process jobs, on-time delivery or to reduce the penalty cost, reduce the workload imbalance and many more. The researches have considered some of the most important objectives in the context of a batch production environment, are

Maximizing the

- Utilization of the systems
- Rate of production

Minimizing the

- Workload imbalance on machines
- Manufacturing lead time
- Penalty cost
- Work-in-process jobs
- Setup and tool changes times

Most of the studies have considered single-criteria algorithms. However, a new scheme of the multi-criteria algorithms has been proposed in recent years. These algorithms do not transform a multi-criteria problem into a single. Simultaneous optimization of all objective functions in the real time environment is challenging and conflicting with each other objective functions. Optimization of any one objective function may not optimize the other objective function. Hence it is necessary to find a better solution that would compromise all the objective functions in the multi objective environment. Literature shows that, researchers have developed various methodologies broadly based on classical mathematical approach, heuristic based and hybrid method to obtain the best solution in the multi-criteria objective environment of the scheduling process. They have studied various traditional, mathematical, heuristic and metaheuristic tools to obtain a reasonably good solution. Many studies have been carried out on the evolutionary approach such as genetic algorithm (GA), ant-colony optimization (ACO), particle swarm optimization (PSO), artificial intelligence etc. Birch [1], H. Nazif [2], Dr. Rakesh Kumar [3], S.Ramya [4], Moin et.al [5] have used the genetic algorithm approach to meet the objective function. The genetic algorithm operating parameters such as population size, selection method, crossover, mutation method and stopping criteria have an effect on the performance of the algorithm in the dynamic situation. The performance of the algorithm depends on the convergence of the solution and the computational time. Simulation has been done in this paper to analyse the effect of crossover probability on objective function such as minimizing the total cost and computational time.In the genetic algorithm, the parent chromosomes are selected from the population to crossover. Darwin's evolution theory states that the best chromosomes should further survive and create new offspring. In the genetic algorithm the selection of parent chromosomes based on different selection operator methods such as a roulette wheel, Boltzman, tournament, rank, steady state, stochastic universal sampling, linear, exponential rank, truncation, and many more [6].

Revised Manuscript Received on October 15, 2019.
Dr. Raghavendra BV, JSS Academy of Technical Education, Bengaluru-560060, India. Email: bvrjss@gmail.com

Crossover is one of the prominent operators used in genetic algorithms. The crossover process is vital in generating new chromosomes by combing two or more parent chromosomes with the hope that they create new and efficient chromosomes. Parent chromosomes are selected based on a selection operator to create the new offspring of the child's chromosomes. During crossover the parent chromosomes are taken in pairs and their genes are exchanged in a certain order, based on the type of crossover method and crossover probability used to obtain the new offspring. It is performed by exchanging alleles between two selected parent chromosomes to explore new solution space. Various crossover operators found in literature are single point crossover, N-point crossover, arithmetic crossover, average crossover, uniform crossover, order crossover, cycle crossover, partially mapped crossover, shuffle crossover, reduced surrogate crossover, flat crossover, discrete crossover, heuristic crossover, statistics-based crossover, adaptive non-uniform crossover [7] and many more. These crossovers have been broadly classified into the classical standard crossover, binary crossover, and application dependant crossover operators. Each crossover operator has its advantages and disadvantages under various circumstances and has an impact on the performance of the genetic algorithm [8, 9, 10, 11, 12].

Goldberg and Lingle have proposed the partially mapped crossover for traveling salesman problem [13]. It is a commonly used crossover operator in permutation encoded chromosomes where the large solution exists. In, Partially Matched Crossover, two crossover points are selected randomly from the parent's chromosomes to produce new offspring's. The crossover process happens through position-by-position exchange operations between the two crossover points in the pair of chromosomes, which gives a matching selection [13, 14]. Partially Mapped Crossover tends to respect the absolute positions [15] and this has been widely used in the scheduling process also.

Crossover probability is the important operator to create the new offspring's from the pair of selected parent's chromosomes. Crossover probability '0' means there is no crossover; offspring is the same as parent's chromosomes from the previous population. Crossover probability 100% means that all offspring is created by crossover. A tradeoff between '0' and 100% probability of crossover would be better to analyze, in hope that the new chromosome population will have both good parts of old chromosomes and maybe the new chromosomes which will converge the solution. It is good to have some parts of chromosomes survive to the next generation of the previous set of population.

Another important operator in the genetic algorithm is the type of mutation used and the probability of mutation. The mutation operator diverges the existing chromosomes and has new chromosomes to avoid the local optimum solution in the genetic algorithm. In the event of no mutation, the offspring will have the same or a copy of the chromosomes as is taken after the crossover process. If the mutation is performed, part of a chromosome is changed depending upon the type of mutation is adopted. The probability of the mutation operator would between 0 and 100 to avoid the local optimum solution and a tradeoff probability will yield a good new offspring's to obtain the best solution. Various types of mutation operators are swap mutation, insert mutation, interchanging mutation, reversing mutation, inversion mutation, creep mutation, scramble mutation, flip mutation, uniform mutation, flip bit, boundary, uniform and many more [16].

Stopping criteria is also a very important parameter in the genetic algorithm. The algorithm should continue to search for convergence solutions and stop if no better solution exists. In the real time dynamic scheduling process the computational time of the algorithm plays a major role. Hence the design of algorithms, GA operating parameters are very important to study and to obtain the best solution within reasonable computational time.

II. PROBLEM FORMULATION AND PROPOSED METHODOLOGY

The problem has been formulated considering the data set for experimentation as mentioned in Table: 1. Experimentation has been carried out on identical parallel machines, having equal capability to process operations and the operation time on identical parallel machines remain the same. The experiment aims to meet the objective function such as minimizing work-in-process material, penalty cost due to nondelivery of the jobs within the specified time, and machine idleness cost due to non availability of processing of the job on the machine in a batch production environment. The fitness function necessary for the genetic algorithm is designed to meet the objective function, and coded in Matlab. The input parameters considered for the experimentation are

Number of parallel machines =6
Number of setups=3
Number of jobs/Part type=10
Batch Quantity for each part type=10
Objective functions: Minimum idleness of the machine and penalty cost.

In the present work genetic algorithm based methodology (Fig: 1) is used to determine the objective function. Initially, randomly generated chromosomes of the population size of 20 are generated. Each chromosome is calculated for the fitness function. Ranking of all the chromosomes has been done and selected first best 12 parents/chromosomes for the next generation and to generate 8 numbers offspring's through crossover and mutation for the next generation. This process is repeated till to the last generation to satisfy the objective function. The scheduling problem is based on NP hard, combinatorial type and hence partially mapped cross operator is used. 150 trials were conducted to decide and found that the algorithm yields convergence of solutions between 26 to 30 generations. Beyond 30 generations no improvement in the solution was found. The experimentation was conducted for various crossover probability of 0.4 to 0.97 and its effect on the performance of the algorithm is studied. 150 iteration (each iteration consists of 30 generations) was conducted for a crossover probability of 0.4 to 0.97 for analyzing the performance of the algorithm

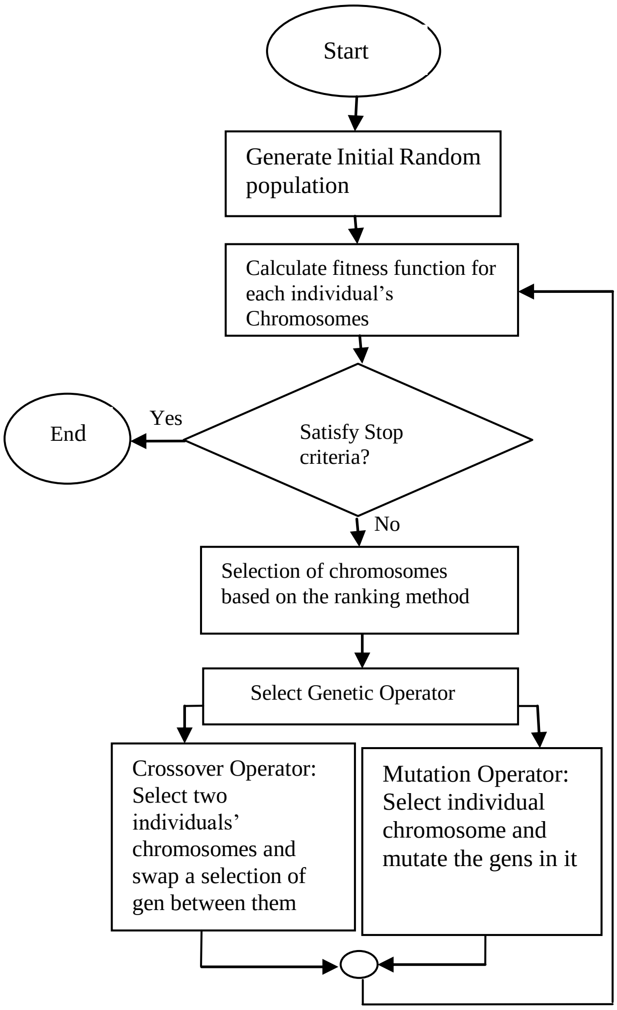

Fig. 1: Genetic Algorithm Flow Chart

Table 1: Data set for Experimentation

Part Type (j)	Operations Time in min			Due Day (dbj)	Penalty Cost Rs./Day/Batch (pbj)
	Set up-I	Set up-II	Set up-III		
1	62	44	2	2	10
2	53	46	-	2	12
3	38	38	20	2	12
4	34	31	10	2	13
5	32	19	10	2	9
6	33	31	9	2	11
7	31	30	16	1	11
8	75	-	-	1	14
9	6.78	-	-	1	8
10	17.34	5.15	15	1	10

Machine Hour rate: Rs. 600 per machine per hour

Table 2: Minimum total cost and average Computational time of 150 iterations for various crossover probability

Crossover Probability	Minimum total cost (Objective Function) in Rs.	Average Computational time of 150 iterations in Second
0.40	4831	3.33
0.45	5118	3.92
0.50	7339	3.31
0.55	3431	3.33
0.60	1670	3.41
0.65	3431	4.07
0.70	5253	3.72
0.75	7040	3.87
0.80	5840	3.87
0.85	5534	3.67
0.90	4649	4.15
0.95	2828	4.34
0.97	9449	4.31

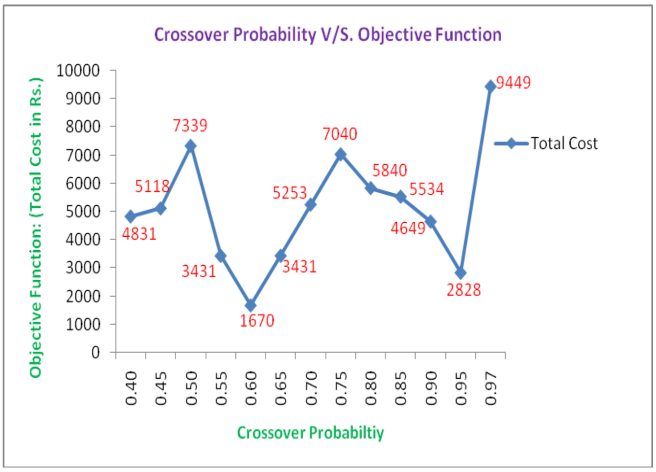

Fig.2: Minimum total cost with respect to various crossover probability

Table 3: Computational time in seconds

Iterations	Crossover Probability			Iterations	Crossover Probability			Iterations	Crossover Probability			Iterations	Crossover Probability		
	0.5	0.55	0.6		0.5	0.55	0.6		0.5	0.55	0.6		0.5	0.55	0.6
1	3.88	3.57	3.30	39	3.16	3.24	3.30	77	3.26	3.37	3.67	115	3.64	3.67	3.53
2	3.33	3.56	3.17	40	3.34	3.29	3.47	78	3.24	3.78	3.50	116	3.20	3.60	3.31
3	3.19	4.10	3.27	41	3.21	3.31	3.55	79	3.31	3.00	3.51	117	3.21	3.36	3.36
4	3.10	3.21	3.47	42	3.28	2.96	3.47	80	3.12	3.17	3.33	118	2.99	3.38	3.19
5	3.29	3.16	3.25	43	3.37	3.31	3.52	81	3.24	3.22	3.27	119	3.17	3.32	3.16
6	3.33	4.02	3.44	44	3.54	3.32	3.50	82	3.34	3.25	3.16	120	3.20	3.18	3.07
7	3.32	3.69	3.25	45	3.20	3.19	3.45	83	3.25	3.34	3.26	121	3.51	3.38	3.10
8	3.36	3.39	3.56	46	3.27	3.26	3.32	84	3.27	3.23	3.30	122	3.51	3.31	3.53
9	3.23	3.51	5.05	47	3.26	3.22	3.59	85	3.34	3.12	3.31	123	3.48	3.20	3.10
10	3.12	3.43	3.45	48	3.27	3.22	3.54	86	3.40	3.28	3.45	124	3.22	3.29	3.39
11	3.24	3.12	3.45	49	3.38	3.05	3.58	87	3.35	3.20	3.14	125	3.26	3.24	3.18
12	3.32	3.29	3.38	50	3.35	3.26	3.85	88	3.21	3.47	3.17	126	3.33	3.37	3.05
13	3.25	3.12	3.36	51	3.52	3.07	3.74	89	3.35	3.46	3.81	127	3.31	3.29	3.22
14	3.29	3.34	3.30	52	3.25	3.27	3.49	90	3.30	3.42	3.47	128	3.12	3.24	3.50
15	3.19	3.44	3.31	53	3.18	3.21	3.45	91	3.17	3.40	3.33	129	3.31	3.38	3.22
16	3.01	3.20	3.39	54	3.48	3.32	3.44	92	3.05	3.15	3.42	130	3.18	3.29	3.28
17	3.22	3.41	3.29	55	3.26	3.27	3.58	93	3.34	3.46	3.37	131	3.08	3.18	3.19
18	2.98	3.39	3.16	56	3.49	3.49	3.40	94	3.32	3.25	3.44	132	3.27	3.57	3.26
19	3.61	3.39	3.17	57	3.23	3.48	3.36	95	3.35	3.59	3.28	133	3.41	3.44	3.40
20	3.40	3.22	3.22	58	3.36	3.16	3.57	96	3.29	3.30	3.40	134	2.91	3.56	3.41
21	3.29	3.40	3.31	59	3.36	3.21	3.63	97	3.11	3.28	3.47	135	3.11	3.36	3.39
22	3.38	3.34	3.24	60	3.38	3.09	3.82	98	3.42	3.24	3.70	136	3.34	3.21	3.26
23	3.39	3.95	3.22	61	3.23	3.18	3.50	99	3.61	3.15	3.26	137	3.36	3.56	3.36
24	3.36	3.31	2.97	62	3.41	3.20	3.60	100	3.08	3.84	3.25	138	3.37	3.28	3.18
25	3.59	3.21	3.42	63	3.27	3.24	3.53	101	4.19	3.76	3.08	139	3.21	3.27	3.43
26	3.00	3.30	3.07	64	3.43	3.21	3.47	102	3.35	2.96	3.28	140	3.11	3.22	3.30
27	3.30	3.46	3.13	65	3.68	3.28	3.30	103	3.38	3.53	3.38	141	3.66	3.48	3.26
28	3.16	3.36	3.42	66	3.35	3.26	3.65	104	3.17	3.41	3.37	142	3.39	3.42	3.39
29	3.34	3.29	3.61	67	3.09	3.24	3.60	105	3.36	3.33	3.33	143	3.08	3.36	3.14
30	3.22	3.23	3.20	68	3.16	3.21	3.52	106	3.19	3.60	3.34	144	3.22	3.20	3.52
31	3.18	3.24	3.42	69	3.28	3.53	3.55	107	3.42	3.53	3.38	145	3.49	3.32	3.25
32	3.38	3.31	3.66	70	3.43	3.34	3.72	108	3.34	3.17	2.92	146	3.21	3.16	3.26
33	3.41	3.25	3.55	71	3.33	3.41	3.65	109	3.20	3.07	3.25	147	3.31	3.38	3.62
34	3.49	3.18	3.79	72	3.39	3.17	3.43	110	3.44	3.40	3.31	148	3.38	3.18	3.47
35	3.39	3.27	3.72	73	3.29	3.44	3.70	111	3.24	3.23	3.32	149	3.31	3.29	3.53
36	3.40	3.29	4.06	74	3.77	3.19	3.46	112	3.12	3.57	3.61	150	3.30	3.23	3.65
37	3.45	3.18	4.25	75	3.47	3.03	3.39	113	3.20	3.47	3.15				
38	3.19	3.26	3.21	76	3.25	3.24	3.39	114	3.27	3.26	3.55				

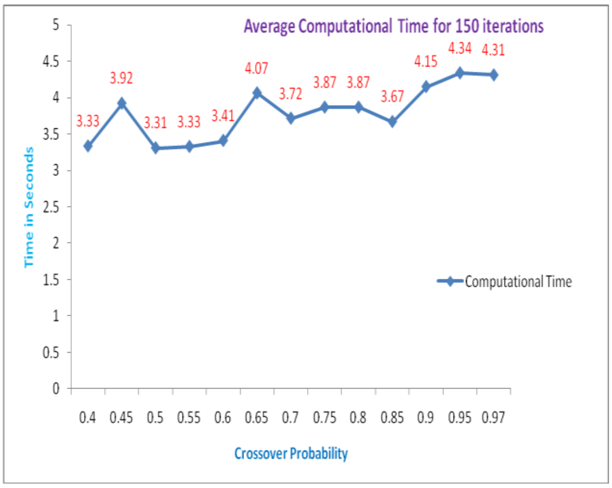

Fig. 3: Average Computational time in seconds of 150 iterations for each crossover probability.

III. CONCLUSION:

The experimentation was conducted on the designed algorithm and coded in Matlab. The genetic algorithm crossover operator- partially mapped crossover method is used for various probabilities of 0.4 to 0.97. The experimental result is presented with the computational time to generate the best possible solution for combined objective functions such as minimizing machine idleness and penalty cost. In the experiment, it is found that the crossover probability of 0.6 converges to the minimum cost of the objective function of Rs. 1670/- (Fig. 2) with the computational time of 3.41 seconds (Fig. 3). The crossover probability of 0.5 gives the minimum computational time of 3.31 seconds but Rs. 7339/- is the generated solution of the objective functions. The second best solution of minimum cost of Rs. 2828/- is at the crossover probability of 0.95 but the computation time is 4.34 seconds (Fig. 3). Experiments for 150 iterations (each iteration of 30 generations) were conducted for the crossover probability rate of 0.4 to 0.97. The minimum computational time for crossover probability 0.5, 0.55 and 0.6 was presented in Table 3. The crossover probability of 0.6 is the converged solution and found reasonably good solution with computational time of 3.41 seconds. This will support the decision maker to obtain good solution at quickly in the real time environment The experiments were conducted on Intel ® Core(TM) I 7-6700 CPU @3.4GHZ. The designed algorithm helps the decision maker to analyze the objective function with the computational time.

REFERENCES

1. Birch, J. B. & Wan, W. An Improved Genetic Algorithm Using a Directional Search. Technical report presented at Virginia Polytechnic Institute and State University, Blacksburg.Year.2009
2. H. Nazif, A Genetic Algorithm on Single Machine Scheduling Problem to Minimise Total Weighted Completion Time. European Journal of Scientific Research, Volume-35, Year: 2009, No.3, pp.444-452.
3. Dr. Rakesh Chawla. Genetic Algorithm approach to Operating system process scheduling problem. International Journal of engineering Science and Technology, Year"2010, pp. 4247-4252.
4. S. Ramya, Window Constrained Scheduling of Processes in Real Time CPU Using Multi Objective Genetic Algorithm "International Journal of Computer Applications: 2010 Volume 1, pp.86-90.
5. Noor Hasnah Moin, Ong Chung Sin, and Mohd Omar, Hybrid Genetic Algorithm with Multiparents Crossover for Job Shop Scheduling Problems, Mathematical Problems in Engineering Volume 2015, pp: 1-12.
6. Firas Alabsi, Reyadh Naoum, Comparison of Selection Methods and Crossover Operations using Steady State Genetic Based Intrusion Detection System, Journal of Emerging Trends in Computing and Information Sciences, Volume. 3, No. 7 July 2012,pp: 1053-1058.
7. Nitasha Soni et al, Study of Various Cross over operators in Genetic Algorithms, International Journal of Computer Science and Information Technologies, Volume. 5 (6) , 2014, pp:7235-7238.
8. A.J. Umbarkar and P.D. Sheth, Crossover operators in Genetic Algorithms: A Review, ICTACT Journal on soft Computing, October 2015, Volume 6, Issue 1, pp:1083-1092..
9. Siew Mooi Lim, Abu Bakar Md. Sultan, Md. Nasir Sulaiman, Aida Mustapha, and K. Y. Leong, Crossover and Mutation Operators of Genetic Algorithms, International Journal of Machine Learning and Computing, Volume. 7, No. 1, February 2017, pp: 09-12.
10. Ellur Anand, Ramasamy Panneerselvam, A Study of Crossover Operators for Genetic Algorithm and Proposal of a New Crossover Operator to Solve Open Shop Scheduling Problem, American Journal of Industrial and Business Management, 2016, 6, pp: 774-789.
11. Padmavathi Kora, Priyanka Yadlapalli, Crossover Operators in Genetic Algorithms: A Review, International Journal of Computer Applications, Volume 162 – No 10, March 2017,pp 34-36
12. D.E.Goldberg. and R.Lingle, "Alleles, loci and the travelling salesman problem", Proceedings of an International Conference on Genetic Algorithms, Morgan Kauffman, 1985, pp 10-19.
13. D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Addison Wesley Longman, Inc., ISBN 0-20115767-5, 1989.
14. S.N. Sivanandam and S. N. Deepa., Introduction to Genetic Algorithms, Springer, ISBN 9783540731894, 2007.
15. Ashish Saxena and Jyotsana panday, Review of Crossover Techniques for Genetic Algorithms International Journal of Trend in Research and Development, Volume 3(5).
16. Nitasha Soni et al, Study of Various Mutation Operators in Genetic Algorithms, International Journal of Computer Science and Information Technologies, Volume. 5 (3) , 2014, pp: 4519-452.

AUTHORS PROFILE

Dr. Raghavendra, BV obtained Ph.D from Visveswaraya Technological University. He is working as Associate Professor in the Department of Mechanical Engineering, at JSS Academy of Technical Education, Bengaluru. He has published more than 15 papers in the field of Scheduling, optimization and composite materials.