Solving Scheduling problems using Selective Breeding Algorithm and Hybrid Algorithm

P.Sriramya[1]
[1]Research Scholar, Department of Computer Science and Engineering, Sathyabama University, Chennai, Tamilnadu, India
Email : sriramya82@yahoo.com

B.Parvathavarthini[2]
[2]Professor & Head, Department of Computer Applications, St. Joseph's College of Engineering, Chennai-119, Tamilnadu, India

M.Chandrasekaran[3]
[3]Department of Mechanical Engineering, Vels University, Chennai, India

Abstract

The n-job, m-machine scheduling problem is one of the general scheduling problems in a system. Scheduling problems vary widely according to specific production tasks but most are NP-hard problems. Scheduling problems are usually solved using heuristics to get optimal or near optimal solutions because problems found in practical applications cannot be solved to optimality using reasonable resources in many cases. In this paper, Selective Breeding Algorithm (SBA) and Hybrid Algorithm (HA) are used for finding optimal for different size benchmark problems. Hybrid Algorithm has Artificial Immune system and shifting bottleneck procedure. The results show that the Selective Breeding algorithm is an efficient and effective algorithm that gives better results than other algorithms compared in literature. The proposed algorithm is a good problem-solving technique for scheduling problems.

Keywords: Scheduling; Benchmark Problems; Selective Breeding Algorithm; Hybrid Algorithm

1. Introduction

Scheduling is defined as the art of assigning resources to tasks in order to insure the termination of these tasks in a reasonable amount of time [1]. The term 'Scheduling' in manufacturing systems is used to the determination of the sequence in which parts are to be processed over the production stages, followed by the determination of the start-time and finish-time of processing of parts, so as to meet an objective or a set of objectives. Also, the problem of scheduling is addressed after the job orders are released into the shop floor, along with their process plans and machine routings [2]. Scheduling plays a crucial role to increase the efficiency and productivity of the manufacturing system. The problem of scheduling is one of the operational issues to be addressed in the system on a daily or weekly basis. Scheduling problems are Non-Polynomial (NP) hard so it is difficult to find optimal solutions [4]. The scheduling can be classified into (i) Single machine scheduling (ii) Flow shop scheduling (iii) Job shop scheduling. The single machine scheduling problem consists of 'n' jobs with same single operation on each of the jobs, while the flow shop scheduling problem consists of 'n' jobs with 'm' operations on each of the jobs. In this problem all the jobs will have the same process sequences. The job shop scheduling problem contains 'n' jobs with 'm' operations on each of the jobs, but in this case the process sequence of the jobs will be different from each other. In general, job scheduling is performed in three stages: short-term, medium-term, and long-term. The activity frequencies of these stages are implied by their names. Long-term job scheduling is done when a new process is created. It initiates processes and so controls the degree of multi-programming. Medium-term scheduling involves suspending or resuming processes by swapping them out of or into memory. Short-term scheduling occurs most frequently and decides which process to execute next [5]. The scheduling is carried out to meet various objectives. These objectives are decided upon the situation, market demands and the customer's satisfaction. There are two broad categories for the scheduling objectives:

- Time based minimization
- Cost based minimization

The objectives considered under the time minimization are Minimize machine idle time, Minimize the mean flow time, Minimize the mean tardiness, Finish each job as soon as possible, Finish the last job as soon as possible. The objectives considered under the Cost minimization are Minimize the costs due to not meeting the due dates, Minimize the maximum lateness of any job, Minimize the total holding cost with no tardy jobs, Minimize the number of late jobs. In this work, the minimization of makespan is considered as objective.

2. Literature Review

The job shop problem is the most complicated and typical problem of all kinds of production scheduling problems, the allocation of resources over time to perform a collection of tasks. Job shop scheduling can be stated as follows: given n jobs that have to be processed on m machines in a prescribed order under certain restrictive assumptions, the objective is to decide how to arrange the processing orders and starting times of operations sharing the same machine. Manufacturing systems with objectives require optimization criteria, such as stock size, due-date reliability and mean lead time.

The job-shop scheduling problem is one of most difficult combinatorial optimization problems [6]. Issues concerning the content and scope of JSPs have been attracting much attention from researchers and practitioners. Optimization and heuristic methods are the two major methods for resolving JSPs. Optimisation methods attempt to find the optimal solution through mathematical programming techniques or methods [8-10]. However, mathematical programming methods are time-consuming, and thus, many researchers focus on developing heuristic algorithms [14-18], algorithms in common use include shifting bottleneck (SB) [19], Tabu search (TS) [13], simulated annealing (SA) [12], the genetic algorithm (GA) [7,11] and artificial immune system (AIS) (22).

3. Problem Description of Scheduling Problem

Normally, the entire job-shop scheduling problem consists of two types of constraints: sequence constraint and resource constraint [23]. The first type states that two operations of a job cannot be processed at the same time. The second type states that no more than one job can be handled on a machine at the same time. Job-shop scheduling can be viewed as an optimization problem, bounded by both sequence and resource constraints. For a job-shop scheduling problem, each job may consist of different number of operations, subjected to some precedence restrictions. Commonly the processing orders of each job by all machines and the processing time of each operation are known and fixed. Once started operations cannot be interrupted. Assume job i(i=1,2,…n) requires processing by machine k(k=1,2,…m) exactly once in its operation sequence (thus, each job has m operations).Let pik is the processing time of job i on machine k, Xik is the starting time of job i on machine k, qik is the indicator which takes on a value of 1 if operation j of job i requires machine k, and zero otherwise. Yihk is the variable which takes on a value of 1 if job precedes job h on machine k, and zero otherwise. The objective function for the given Job Shop Scheduling is

Minimize Z = Cmax

Subject to

(i) Sequence constraint

ie., for a given job i, the (j+1)st operation may not start before the jth operation is completed.

(ii) Resource constraint

ie. Only one job will be processed in a machine.

4. Numerical Illustration of scheduling Problem

LA16 Benchmark Problem [20]

Number of machines: 10 Number of jobs: 10

Operational sequences and corresponding processing times of LA 16 problem is shown in Table 1 & 2.

Table 1. Operation Routing

1	6	9	8	7	2	0	4	3	5
4	2	5	9	0	7	1	8	6	3
3	2	8	1	4	9	7	6	0	5
1	3	2	7	8	9	6	0	5	4
2	0	5	6	7	1	4	9	3	8
2	3	5	9	4	6	0	8	1	7
3	2	0	1	9	8	6	5	4	7
1	0	3	4	6	9	8	5	2	7
4	2	8	5	3	7	1	6	9	0
8	9	2	4	3	0	7	6	1	5

Table 2. Processing Time

21	71	16	52	26	34	53	21	55	95
55	31	98	79	12	66	42	77	77	39
34	64	62	19	92	79	43	54	83	37
87	69	87	38	24	83	41	93	77	60
98	44	25	75	43	49	96	77	17	79
35	76	28	10	61	9	95	35	7	95
16	59	46	91	43	50	52	59	28	27
45	87	41	20	54	43	14	9	39	71
33	37	66	33	26	8	28	89	42	78
69	81	94	96	27	69	45	78	74	84

Step 1: Initial population (10 job sequences) is generated.

Each seed is operated 10 times for achieving one complete job sequence.

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

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

Step 2: Calculation of makespan and breeding factor for each sequence.

i) Breeding factor = 1/ makespan

ii) Sorted Sequences in descending order based on its breeding factor

Step 3: Divide the population set into exactly two sets and form diploid cells.

In dominant set R contains 5 job sequences namely R1, R2, R3, R4 and R5. In recessive set r contains 5 job sequences namely r1, r2, r3, r4 and r5. Form the diploid set such as {R1r1, R2r2, R3r3, R4r4, and R5r5}.

Dominant set (R)	Makespan	Recessive set (r)	Makespan
9- 10- 6- 3- 7- 1- 2- 4- 5- 8 =>	970	3- 9- 2- 7- 5- 10- 6- 8- 1- 4 =>	1087
1- 10- 6- 7- 5- 8- 9- 3- 2- 4 =>	1009	1- 4- 5- 10- 8- 6- 9- 3- 2- 7 =>	1114
3- 9- 1- 6- 10- 8- 5- 7- 2- 4 =>	1079	6- 5- 3- 9- 7- 1- 2- 8- 10- 4 =>	1115
3- 1- 2- 7- 9- 10- 8- 6- 5- 4 =>	1079	9- 4- 6- 8- 10- 3- 7- 1- 5- 2 =>	1135
3- 1- 10- 8- 5- 6- 9- 7- 2- 4 =>	1079	9- 6- 1- 4- 7- 3- 10- 2- 8- 5 =>	1198

Step 4: Breeding process

Possible breeds for one set combination by considering two diploids namely R1r1 and R2r2.

	R2	r2
R1	**R1R2**	**R1r2**
r1	**r1R2**	**r1r2**

Find all the possible breeds (diploid set)**.**

R1R2 => 9- 10- 6- 3- 7- 1- 2- 4- 5- 8 **&** 1- 10- 6- 7- 5- 8- 9- 3- 2- 4
R1r2 => 9- 10- 6- 3- 7- 1- 2- 4- 5- 8 **&** 1- 4- 5- 10- 8- 6- 9- 3- 2- 7
r1R2 => 3- 9- 2- 7- 5- 10- 6- 8- 1- 4 **&** 1- 10- 6- 7- 5- 8- 9- 3- 2- 4
r1r2 => 3- 9- 2- 7- 5- 10- 6- 8- 1- 4 **&** 1- 4- 5- 10- 8- 6- 9- 3- 2- 7

Similarly form diploid sets for all possible combinations.

Step 5: Fusion process

Fusion process for all the possible diploid combinations are done and separated as haploid. For each set, select fusion points.

Number of fusion points = length of the given haploid / 2.

Fusion points are selected randomly.

At the fusion points interchange positions between parents.

Consider one set of diploid: Fusion points are 1, 2, 4, 6, and 9

Original sequence	After Fusion
9- 10- 6- 3- 7- 1- 2- 4- 5- 8	**1- 10**- 6- **7**- 7- **8-** 2- 4- **2**- 8
1- 10- 6- 7- 5- 8- 9- 3- 2- 4	**9- 10**- 6- **3**- 5- **1**- 9- 3- **5**- 4

Step 6: Inbreeding Depression Process

Selective breeding of particular genes runs the risk of losing some of the other genes from the gene pool altogether, which is irreversible. This is called inbreeding depression. To avoid this, add 10% newly generated haploids in each iteration. This gives a chance to search solution in new search space.

Step 7: Sort the final Sequences based on breeding factor

First ten sequences are used for next iteration which is given below.

Sequence	Makespan	Sequence	Makespan
9- 10- 6- 1- 7- 8- 2- 3- 5- 4	970	9- 10- 6- 7- 4- 1- 8- 2- 5- 3	1008
9- 10- 6- 3- 7- 1- 2- 4- 5- 8	970	3- 9- 5- 7- 8- 10- 6- 2- 1- 4	1009
9- 10- 6- 3- 7- 1- 4- 2- 8- 5	971	1- 5- 3- 7- 9- 8- 2- 6- 10- 4	1009
9- 10- 6- 8- 4- 1- 7- 2- 5- 3	978	1- 9- 7- 10- 8- 6- 5- 3- 2- 4	1009
6- 9- 3- 7- 1- 10- 2- 8- 5- 4	980	1- 10- 6- 7- 5- 9- 8- 3- 2- 4	1009

After completing 100 iterations, the result is

2- 10- 9- 8- 7- 3- 4- 6- 5- 1 -> Makespan = 945

10- 5- 9- 3- 4- 6- 8- 2- 1- 7 -> Makespan = 945

The SBA algorithm is implemented in C language on personal computer Pentium IV 2.4 GHz. The maximum number of iterations has been set to 100 X n, where n is the number of jobs

5. Results and Discussion

Selective Breeding algorithm (SBA) has been tested for six problem instances of various sizes collected in the following classes: Three instances denoted as (ORB2, ORB4, and ORB5) and two instances denoted as (ABZ5, ABZ7) and LA16 instance. The Relative Error RE (%) was calculated for all problem instances, as a percentage by which the solution obtained is above the optimum value (Opt) if it is known or best known lower bound (LB) [19].

$$RE (\%) = 100 X (UB – LB)/LB.$$

In Table , the solutions for class (A) problems obtained from SBA is compared with Artificial Immune System and Hybrid Algorithm. SBA gives optimum bound value for 6 out of 6 problems where as AIS gives optimum value for 3 out of 6 problems hybrid gives optimum bound value for 2 out of 6 problems. The mean relative error is zero in SBA which is lower than previously obtained value of 1.97 % of AIS and 3.148 % of hybrid algorithm

Table 3. Results obtained in SBA

Problem	Jobs	Machines	Optimal Value	**SBA**	**RE_{SBA}**	AIS	RE_{AIS}	Hybrid Algorithm	RE_{HA}
LA16	10	10	945	**945**	**0**	945	0	955	1.05
ORB 2	10	10	888	**888**	**0**	891	0.34	891	0.34
ORB4	10	10	1005	**1005**	**0**	1005	0	1005	0
ORB5	10	10	887	**887**	**0**	888	0.11	889	0.23
ABZ5	10	10	1234	**1234**	**0**	1234	0	1234	0
ABZ7	20	15	661	**661**	**0**	666	1.52	666	1.52
Mean Relative Error (MRE)					**0**		1.97		3.148

6. Conclusion

In this paper, the proposed Selective Breeding approach has been used for solving job shop scheduling problem with the objective of makespan minimization. The Selective Breeding approach uses simple but effective techniques for calculating breeding process, fusion process and Inbreeding depression process. The algorithm has been tested on six benchmark problem instances. The findings were compared with Artificial

Immune System and hybrid algorithm. The proposed Selective Breeding approach found optimal results for all the tested problem instances**.**

References

[1] Bruker P. (1995), Scheduling Algorithms, 2nd Edn, Springer-Verlag, Berlin.

[2] French S (1982), Sequencing and scheduling: An introduction to the mathematics of the job shop, New York, Wiley.

[3] Yang S., Dingwei Wang (2001) A new adaptive neural network and heuristics hybrid approach for job shop scheduling, Computers and operations Research, 28, pp 955-971.

[4] Garey M et al (1976), The complexity of flow shop and job shop scheduling, Mathematics of Operations Research, 1, pp 117-129.

[5] Panneerselvam R, Production and operations Management, PHI Learning Pvt. Ltd

[6] Baker K. R. (1974), Introduction to Sequencing and Scheduling, John Wiley, New York.

[7] Byung Joo Park, Hyung Rim Choi, Hyun Soo Kim (2003), A hybrid genetic algorithm for the job shop scheduling problems, Computers & Industrial Engineering, 45, pp 597–613.

[8] Erschler JF, Roubellat JP, Vernhes, (1976), Finding some essential characteristics of the feasible solutions for a scheduling problem. Operations Research, 24, pp 774- 783.

[9] Dubois D, Fargier H, Prade H., (1995), Fuzzy constraints in job shop scheduling. Journal of Intelligent Manufacturing, 6, pp 215-234.

[10] Montazeri H., (1990), Analysis of scheduling rules for an FMS. International Journal of Production Research, 28, pp 785- 802.

[11] Croce F.D., Tadei.R and Volta.G, (1995), A Genetic algorithm for the job shop problem, Computers and Operations Research, 22, pp 12-24.

[12] Ponnambalam S.G., Jawahar.N and Aravindan.P, (1999), A simulated Annealing algorithm for job shop scheduling, Production Planning and Control, 10(8), pp 767-777.

[13] Ponnambalam S.G., Aravindan.P & Rajesh S.V(2000), A Tabu Search algorithm for job shop scheduling, International Journal of Advanced Manufacturing Technology, 16, pp 765-771.

[14] Christos A.Kaskavelis, Michaek C.Caramanis, (1998), Efficient Langrangian relaxation algorithms for industry size job shop scheduling problems, IIE tranactions, Vol 30, pp 1085-1097.

[15] Binato.s, Hery W.J., Loewenstern, and Resende M.G.C, (2000), A Grasp for job shop scheduling, AT&T Labs Research Technical Report: 00.6.1.

[16] Khaled Mesghouni, Silm Hammadi and Pierre Borne, (2004), Evolutionary algorithms for job shop scheduling, International Journal of Application of Mathematics and Computer Science, Vol 14, pp 91-103.

[17] http://www.biology-online.org/2/12_selective_breeding.html

[18] Applegate D., Cook W., (1991), A computational study of the job shop scheduling problem, ORSA Journal on Computing 3, pp 149-156.

[19] Adams J., Balas E., Zawack D., (1988), The shifting bottleneck procedure for job shop scheduling, Management Science, 34, pp 391-401.

[20] Lawrence S., (1984), Supplement to Resource constrained project scheduling: an experimental investigation of heuristic scheduling techniques, Tech. Rep., GSIA, Carnegie Mellon University.

[21] Taillard E., (1993), Benchmarks for basic scheduling problems, European Journal of Operational Research, 64, pp 278-285.

[22] Chandrasekaran M., Asokan P., Kumanan S., Balamurugan T., (2006), Solving job shop scheduling problem using Artificial Immune System, International journal of Advanced Manufacturing Technology, Springerlink, December 2006, Volume 31, Issue 5-6, pp 580-593

[23] Yang, S., & Wang, D. (2001). A new adaptive neural network and heuristics hybrid approach for job-shop scheduling. Computers & Operation Research, 28(10), 955–971.

[24] Sriramya P, Parvathavarthini B, Balamurugan T, (2013), A novel evolutionary selective breeding Algorithm and its application, Asian Journal of Scientific Research, 6: 107-114.