Indian Journal of Engineering

ANALYSIS

International Bi-annual Journal for Engineering
ISSN 2319 – 7757 EISSN 2319 –7765

A Heuristic Genetic Approach Applied To Lawrence Problems of Job Shop Scheduling For Minimization of Makespan

Udaiyakumarr KC[1]*, Chandrasekaran M[2]

1.Research Scholar, Sathyabama University, Chennai, Taminnadu, India; E-mail: kcudaiyakumar1967@gmail.com
2.Director, Department of Mechanical Engineering, Vel's University, Chennai, Tamilnadu, India

***Correspondence**: K.C.Udaiyakumar; Assistant Professor(S.G), Department of Mechanical Engineering, SRM University, Ramapuram, Chennai – 600089; E-mail: kcudaiyakumar1967@gmail.com

Publication History
Received: 24 September 2014
Accepted: 27 October 2014
Published: 19 November 2014

Citation
Udaiyakumarr KC, Chandrasekaran M. A Heuristic Genetic Approach Applied To Lawrence Problems of Job Shop Scheduling For Minimization of Makespan. *Indian Journal of Engineering*, 2014, 11(27), 41-46

ABSTRACT

Last few decades the problem of scheduling several jobs on number of machine with each job having specific machine route has been researched widely. This job shop problem is one of the most well-known hardest combinatorial or non polynomial hard optimization problem. This lead to recent interest in using genetic algorithm (GA). How to adapt genetic algorithm to the job-shop problem is extremely challenging work. Many research has done in encoding a solution into a chromosome and improving the performance of genetic search. The problem explored in this research revolves around the allocation of operation to the machine and sequencing of operations on machine under sequence constraint of job pre-specified. It is very hard to integrate both of the above due to conflicting objectives or inability to communicate the dynamic changes in the job shop. In this paper genetic algorithm work is applied on Lawrence (LA01-40) problems to find minimization of makespan. Also, to compare this genetic algorithm results with the benchmark values and existing heuristic. The numerical example showed good result.

Keywords: Genetic algorithm, benchmark, heuristic, makespan, optimization.

1. INTRODUCTION

The job shop scheduling problem also referred to as the JSP, can be described by a set of n jobs Ji, $_{(1<i<n)}$ which is to be processed on a set of 'm' machines. The problem can be characterized as follows:

1. Each job must be processed on each machine in the order given in a pre-defined sequence of machines.
2. Each machine can process only one job at a time.
3. The processing of job 'Jj' on machine 'Mr' is called the operation 'Ojr'.

4. Operation 'Ojr' requires the exclusive use of 'Mr' for an uninterrupted duration 'Pjr', its processing time.
5. The starting time and the completion time of an operation 'Ojr' is denoted as 'Sjr' and 'Cjr' respectively.
6. The time required to complete all the jobs is called the makespan, which is denoted as 'Cmax'.

The technological sequence of machines can be different for each job as implied in the first condition and that the order of jobs to be processed on a machine can be also different for each machine.

The objective of optimizing the problem is to find a schedule that minimizes Cmax. . The Gantt chart is a convenient way of visually representing a solution of the JSP. The Gantt chart shows time units at the abscissa and machine numbers at the axis of ordinate for easy understanding. Computational time and the complexity increases with increase in the size of a scheduling problem. Resources (where) operate on tasks (what) for specific periods of time (when). Using this simple classification, and neglecting precedence and other constraints. At certain times or within certain time limits, a feasible solution is not guaranteed. Some algorithms are capable of determining if such an infeasible situation exists. Most heuristic methods cannot.

Many job shop, flow shop, and production scheduling problems are available and have been described in the literature(Beasley). The classic job shop problems include the 6x6 and 10x10 problems first presented in the year 1963(Muth and Thompson, 2003). Since then, many others have suggested and solved a variety of more complex variations. A set of problems was posted on the world-wide web in the early 1995 (Fox, 1996).

Heuristic Solution Methods, generates exact solution methods are guaranteed to find the optimal solution (if one exists), heuristic methods sometimes find optimal solutions, but more often find simply good solutions. Heuristic methods typically require far less time and/or space than exact methods. Artificial Intelligence Approaches, Hildum grouped artificial intelligence approaches to scheduling as either expert systems or knowledge-based. Both are structured heuristic methods that differ in the way they control the application of their application-specific heuristics.. The rule base is tailored explicitly to a specific problem, so expert systems are typically highly specialized. Knowledge-based systems typically split the problem into sub-problems or different views. Dorndorf et. al. (2002) used a genetic algorithm to find optimal sequences of local decision rules to be used with traditional search algorithms for a range of static, deterministic job shop scheduling problems. Their method found shorter make spans more quickly than the shifting bottleneck procedure of peter et. al., (1992) or simulated annealing method (Yamada, 2003).

2. GENETIC ALGORITHM

One of the earliest suggested uses of genetic algorithms for scheduling was attempted by Lawrence Davis. GA is based on Darvin's Theory. This has been used to evolve heuristics for scheduling. Genetic Algorithms (GAs)(Jain and Meeran, 1999) are search strategies designed after the mechanics of natural selection and natural genetics to optimize highly complex objective functions. GAs have been quite successfully applied to various optimization problems including scheduling. Genetic Algorithms use a ward borrowed from natural genetics. A set of individuals called population. An individual has two representations called phenotype and genotype. The phenotype represents a potential solution to the problem to be optimized in a straightforward way used in the original formulation of the problem. The genotype, on the other hand, gives an encoded representation of a potential. A chromosome is made of genes arranged in linear succession. For example, a chromosome consists of a sequence of 0 or 1 (i.e. a bit string), .An optimization objective function is expressed as

$F(x) = \max f(x)$ (Eqn.1)

where x is an individual in the current population P**.**

3. ENCODING SCHEME

For encoding all Lawrence series benchmark problems taken from OR Library (Beasley) for consideration, here Lawrenceare 01 problem encoded is given in the figure.1

Lawrence problem 1 : Lawrence 10x5 instance
1 21 0 53 4 95 3 55 2 34
0 21 3 52 4 16 2 26 1 71
3 39 4 98 1 42 2 31 0 12
1 77 0 55 4 79 2 66 3 77
0 83 3 34 2 64 1 19 4 37
1 54 2 43 4 79 0 92 3 62
3 69 4 77 1 87 2 87 0 93
2 38 0 60 1 41 3 24 4 83
3 17 1 49 4 25 0 44 2 98
4 77 3 79 2 43 1 75 0 96

Fig. 1. Encoding of Lawrence problem (LA 01)

Udaiyakumarr and Chandrasekaran,
A Heuristic Genetic Approach Applied To Lawrence Problems Of Job Shop Scheduling For Minimization Of Makespan,
Indian journal of engineering, 2014, 11(27), 41-46,
http://www.discovery.org.in/ije.htm

4. POPULATION GENERATION

In this stage 'n' number of parents with 'm' number of strings were generated randomly. Normally better solution can be obtained as the population size increases, but proportionally computational time increases and a trade-off exist between the both. In this proposed research, population size was set to minimum of 10 with variable length strings in a parent were generated using a randomize function. The string length is decided from the table 1

Table 1

Condition for Fixing Parent Size

Problem	Problem Size	No. of strings
Lawrence 01 – 05	10 x 5	50
Lawrence 06 – 10	15 X 5	75

5. EXPERIMENTAL WORK

By conducting various experiments, it was found that the population size of 2 leads to more number of generations and the result doesn't give much difference beyond the population size of 100**.** Moreover, most of the researchers set the population size as 10. The parent size depends on the problem size. Various experiments were conducted for the parent size of 2 to 100. The parent size of 10 and 15 produces better results in lesser computational time. Therefore, in the proposed work the parent size was set to 10. The visual basic code used for generating the random population are given in the figure .2.

VISUAL BASIC CODE

```
For j = 1 To 10 (no of parents)
  For i = 1 To 15 (string length)
    Randomize
    Genome (j) = Genome (j) & Int ((max - min + 1) * Rnd + 1)
    End If
    Next i
  Next j
```

Figure.2. Random Parent Generation Code

The sample of 10 set of parents generated using the random function in this work is given in the figure 3.

Parent	16132018171219151114
1	-19131820111412161715
	-15161312191114201718
	-14151719131611201218
	-13161418171211192015 -
Parent 2	14201611131718191512
	-15181917121411201316
	-14201917181311121615
	-12141615172011181319
	-17131220191611181415 -
Parent 3	13121416201817191511
	-15141913182011121617
	-11201714151613121918
	-14151719131611201218
	-13161418171211192015 -
Parent 4	14201611131718191512
	-15181917121411201316
	-14201917181311121615

Udaiyakumarr and Chandrasekaran,
A Heuristic Genetic Approach Applied To Lawrence Problems Of Job Shop Scheduling For Minimization Of Makespan,
Indian journal of engineering, 2014, 11(27), 41-46,
http://www.discovery.org.in/ije.htm

-12141615172011181319

-17131220191611181415 -

Parent 5 13121416201817191511

-15141913182011121617

-11201714151613121918

-14151719131611201218

-13161418171211192015 -

Parent 6 14201611131718191512

-15181917121411201316

-14201917181311121615

-12141615172011181319

-17131220191611181415 -

Parent 7 13121416201817191511

-15141913182011121617

-11201714151613121918

-14151719131611201218

-13161418171211192015 -

Parent 8 14201611131718191512

-15181917121411201316

-14201917181311121615

-12141615172011181319

-17131220191611181415 -

Parent 9 13121416201817191511

-15141913182011121617

-11201714151613121918

-14151719131611201218

-13161418171211192015 -

Parent 10 14201611131718191512

-15181917121411201316

-14201917181311121615

-12141615172011181319

-17131220191611181415 –

6. FITNESS FUNCTION

Fitness function is also called as the objective function, which is used to select the best parent from the generated population. The developed fitness function F(x) is given in equation 2.

Min. $F(x) = Max.(P_{time})$ For all machines, sequences Eqn.2

By the use of GA on all LA 01-40 problems(Beasley), the makespan values are determined and shown in table 2.and also compared with benchmark values(Taillard, 1993; Jain and Meeran, 1999) and their work computational time also shown .

Table 2

Comparison of GA Results of Lawrence 01-40 series with bench mark values[and computational time .

Law problem	N	M	Makespan obtained by GA	Optimum Makespan	Relative Error (%)	Compt. Time
1	10	5	666	666	0	9.8608
2	10	5	658	655	0.458	9.9113
3	10	5	597	597	0	8.4278
4	10	5	614	590	4.06	9.4298
5	10	5	593	593	0	9.1878
6	15	5	932	926	1.59	12.590
7	15	5	890	890	0	11.087
8	15	5	863	863	0	12.517
9	15	5	962	951	1.156	13.645
10	15	5	958	958	0	12.218
11	20	5	1222	1222	0	24.259
12	20	5	1039	1039	0	21.844
13	20	5	1150	1150	0	22.000
14	20	5	1295	1292	0.232	23.686
15	20	5	1207	1207	0	22.845
16	10	10	945	945	0	26.265
17	10	10	784	784	0	27.181
18	10	10	848	848	0	25.419
19	10	10	842	842	0	26.390
20	10	10	902	902	0	26.098
21	15	10	1046	1046	0	62.159
22	15	10	929	927	0.216	61.685
23	15	10	1032	1032	0	62.162
24	15	10	935	935	0	50.525
25	15	10	977	977	0	61.466
26	20	10	1218	1218	0	101.29
27	20	10	1235	1235	0	101.77
28	20	10	1225	1216	0.740	110.13
29	20	10	1142	1142	0	100.89
30	20	10	1355	1355	0	114.13
31	30	10	1810	1784	1.457	116.49
32	30	10	1850	1850	0	120.64
33	30	10	1719	1719	0	119.86
34	30	10	1721	1721	0	120.20
35	30	10	1888	1888	0	122.50
36	15	15	1268	1268	0	126.04

Udaiyakumarr and Chandrasekaran,
A Heuristic Genetic Approach Applied To Lawrence Problems Of Job Shop Scheduling For Minimization Of Makespan,
Indian journal of engineering, 2014, 11(27), 41-46,
http://www.discovery.org.in/ije.htm

37	15	15	1421	1397	1.717	125.05
38	15	15	1211	1196	1.254	121.68
39	15	15	1233	1233	0	126.75
40	15	15	1258	1222	2.946	129.66
Mean Relative Error %					0.395	

7. CONCLUSION

In this work, GA has been developed to solve single-objective minimization of makespan Lawrence 1-40 series taken from OR Library and solved. The genetic parameters have been set by conducting various test cases and the genetic operators have also been modified to suit the need. Crossover operator has been enhanced to to improve the convergence rate and thereby reducing the computational time. It was found that the performance were better compared to the steady state GA..Mutation operator used to avoid stagnation and to check the probability of various patterns. Mutation operator has been developed to change the gene at random, thereby it avoids the stagnation, increases the search space and also identifies the best job. Swapping operator has been applied to increase the search space for identifying the better solution. This work is not to prove that the genetic algorithm is better. But, it proved that the genetic algorithm can be an effective algorithm for JSP with related constraints.

REFERENCES

1. Muth, Thompson. A Promising genetic Algorithm Approach to Job-Shop Scheduling, ,Rescheduling, and Open-Shop Scheduling Problems. *Proceedings of the Fifth International Conference on Genetic Algorithms, S. Forrest (ed.), San Mateo: Morgan Kaufmann*, 2003, 375-382.
2. Fox B. An algorithm for scheduling improvement by scheduling shifting. *Technical report 96.5.1, Mcdonnel douglas aerospace – Housto,* 1996.
3. Dorndorf U, Pesch E, Phan-Huy T. Constraint propagation and problem decomposition : A preprocessor procedure for the job shop problem. *Ann. Oper. Res*, 2002, 115, 125 – 145.
4. Peter JM, van Laarhoven, Emile HL, Aarts, Jan Karel Lenstra. Job Shop Scheduling by Simulated Annealing. *Operations Research*, 1992, 40(1), 113-125
5. Yamada T. Studies on Metaheuristics for Jobshop and Flowshop Scheduling Problems in Department of Applied Mathematics and Physics. vol. Doctor of Informatics Kyoto, Japan: Kyoto University 2003, 120
6. Beasley JE. Job shop scheduling, *OR Library.*
7. Taillard E. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 1993, 64(2), 278-285
8. Jain AS, Meeran S. Deterministic job-shop scheduling: Past, present and future. *European journal of operational research*, 1999, 113(2), 390-434.