A Genetic Algorithm for Job-Shop Scheduling

Ye LI
Transportation Management College, Dalian Maritime University, Dalian, PR.China
liye_dlmu@sohu.com

Yan CHEN
Transportation Management College, Dalian Maritime University, Dalian, PR.China
Chenyan_dlmu@163.com

***Abstract*—In this paper, we analyze the characteristics of the job shop scheduling problem. A new genetic algorithm for solving the agile job shop scheduling is presented to solve the job shop scheduling problem. Initial population is generated randomly. Two-row chromosome structure is adopted based on working procedure and machine distribution. The relevant crossover and mutation operation is also designed. It jumped from the local optimal solution, and the search area of solution is improved. Finally, the algorithm is tested on instances of 8 working procedure and 5 machines. The result shows that the genetic algorithm has been successfully applied to the job shop scheduling problems efficiency.**

***Index Terms*—job shop scheduling, genetic algorithm, initial population, crossover and mutation operation**

I. INTRODUCTION

Job-shop scheduling is usually a strongly NP-complete problem of combinatorial optimization problems and is the most typical one of the production scheduling problems. Since it is an important practical problem, some researchers have formulated various JSP models based on different production situations and problem assumptions[1,2].

Dessouky and Leachman developed two integer programming formulations which could easily handle high-volume manufacturing such as multiple machines of the same type, demand size greater than one unit for a particular product type and repeat visit to the same machine type[3]. Makoto and Hiroshi considered the JSP problem to minimize the total weighted tardiness with job-specific due dates and delay penalties, and a heuristic algorithm based on the tree search procedure was developed for solving the problem[4]. Gomes and Barbosa presented an integer linear programming model to schedule flexible job shop, which considered job re-circulation and parallel homogeneous machines[5]. Loukit and Jacques dealt with a production scheduling problem in a flexible job shop with particular constraints-batch production[6]. Jason presented mix integer linear programming, which considered job re-circulation. The objective was to minimize the completion time[7]. Liu regarded the JSP problem with dynamic shop scheduling problem[8]. Borstjan and Peter proposed an alternative way to avoid infeasibility by incorporating a repairing technique into the mechanism for applying moves to a schedule. [9]. Hiroshi, Toshihiro considered the job shop scheduling problem of minimizing the total holding cost of completed and in-process products subject to no tardy jobs [10].

Since Davis (1985) proposed the first GA-based technique to solve scheduling problem, GA has been used with increasing frequency to address scheduling problems. The GA utilizes a population of solution in its search, giving it more resistance to premature convergence on local minima[11,12]. Hong Zhou and Yuncheng Feng proposed a hybrid heuristics GA for $n/m/G/C_{\max}$, where the scheduling rules, such as shortest processing time(SPT) and MWKR, were integrated into the process of genetic evolution[13]. Byung developed an efficient method based on genetic algorithm to address JSP. The scheduling method based on single genetic algorithm and parallel genetic algorithm was designed[14]. Dirk and Christian considered a job shop scheduling problems with release and due-dates, as well as various tardiness objectives. The genetic algorithm can be applied to solve this kind of problem[15].

In this paper, a university mathematical model for agile job-shop scheduling problem is constructed. The objective of this model is to minimize makespan. In order to solve this mixed- and multi-product scheduling problem, a new genetic optimization process based on GA will be developed, which includes initialization population and two kinds of coding and crossover and mutation operators based on work procedure and machine distribution. The neighborhood structure is extended and the globally optimal solution is obtained. The algorithm will then be used to solve the JSP problem of 8 working procedure and 5 machines. The Gantt chart of processing route is draw, and minimizes the completion time of all the jobs.

II. MODELING THE JOB-SHOP SCHEDULING PROBLEM

The job shop scheduling problem is a generalization of the classical job shop problem. In the static job-shop scheduling problem, finite jobs are to be processed by finite machines. Each job consists of a predetermined sequence of task operations, each of which needs to be processed without preemption for a given period of time on a given machine. Tasks of the same job cannot be processed concurrently and each job must visit each machine exactly once. Each operation cannot be

Manuscript received July 1, 2009; revised Augest 30, 2009; accepted September 4, 2009.

doi:10.4304/jsw.5.3.269-274

commenced until the processing is completed, if the precedent operation is still being processed. A schedule is an assignment of operations to time slots on a machine. The makespan is the maximum completion time of the jobs and the objective of the JSSP is to find a schedule that minimizes the makespan.

We consider the flexible case where stages might be skipped. Every job is a chain of operations and every operation has to be processed on a given machine for a given time. The task is to find the completion time of the very last operation is minimal. The chain order of each job has to be maintained and each machine can only process one job at the same time. No job can be preempted; once an operation starts it must be completed; two operations of a job can not be processed at the same time; no more than one job can be handled on a machine at the same time; the same priority level at each operation; there is no setup and idle time; there is no break time; all machines are available at zero in the usage time; machine efficiency is 100%; the money value is not considered. The following additional definitions and notation will help in formulating the problem:

(1) i : number of machines;

(2) j_i : number of operations of machine i ;

(3) p_{ij} : processing time of operation j on machine i ;

(4) o_j : sequence and technique restriction of job;

(5) t_{ij} : starting time of operation j on machine i ;

(6) t_j : completion time of operation j .

(7) $X_{ijk} = \begin{cases} 1 & \text{if operation } j \text{ precedes operation} \\ 0 & \text{otherwise} \end{cases}$

(8) $z_{ij} = \begin{cases} 1 & \text{if operation } j \text{ is allocated on machine} \\ 0 & \text{otherwise} \end{cases}$

(9) $C_{\max}$: makespan

According to above suggestion, parameter and decision variable of problem, the mathematical model is identified as followed:

$$\min C_{\max}$$

s.t

$$\sum_{j=1}^{n} z_{ij} = 1 \qquad t_{ij} + p_{ij} \le t_{ik} + (1 - X_{ijk})M$$

$$t_{ij} + p_{ij} \le t_{i+1,j} \sum_{j=1, j\neq k}^{n} \sum_{i=1}^{m} X_{ijk} = 1$$

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ijk} + X_{ikj}) \le 1$$

III. GA FOR JOB SHOP SCHEDULING PROBLEM

The GA was first introduced by Holland(1975). It is a stochastic heuristics, which encompass semi-random search method whose mechanism is based on the simplifications of evolutionary process observed in nature. As opposed to many other optimization methods, GA works with a population of solutions instead of just a single solution. GA assigns a value to each individual in the population according to a problem-specific objective function. A survival-of-the-fittest step selects individuals from the old population. A reproduction step applies operators such as crossover or mutation to those individuals to produce a new population that is fitter than the previous one. GA is an optimization method of searching based on evolutionary process. In applying GA, we have to analyze specific properties of problems and decide on a proper representation, an objective function, and a construction method of initial population, a genetic operator and a genetic parameter. The following sub-sections describe in detail how the GA is developed to solve the above JSP problem.

A. Chromosome Representation and Decoding

The first step in constructing the GA is to define an appropriate genetic representation (coding). A good representation is crucial because it significantly affects all the subsequent steps of the GA. Many representations for the JSP problem have been develop.

In this study, two representations based on working sequence and machine distribution are constructed. If the number of the machine type t(t>1), the genes in each chromosome will be divided into t parts in turn. Each part represents one type of machine. Each operation can only be assigned to the machines which can handle it. For example, suppose a chromosome is given as [4221134233114234] in 4 job×4 machines problem. Here, 1 implies operation of job J_1, and 2 implies operation of job J_2, and 3 implies operation of job J_3, and 4 implies operation of job J_4. Because there are four operations in each job, it appears the four times in a chromosome. Such as number 2 being repeated the forth in a chromosome, it implies four operations of job J_2. The first number 2 represents the first operation of job J_2 which processes on the machine 1. The second number 2 represents the second operation of job J_2 which processes on the machine 2, and so on. The representation for such problem is based on two-row structure, as following:

Chromosome gene	4	2	2	1	1	3	4	2	3	3	1	1	4	2	3	4
job-operation	4-1	2-1	2-2	1-1	1-2	3-1	4-2	2-3	3-2	3-3	1-3	1-4	4-3	2-4	3-4	4-4

Figure 1. Chromosome Genes and Operations

Chromosome gene	4	2	2	1	1	3	4	2	3	3	1	1	4	2	3	4
Operation-machine	1-1	1-2	1-3	1-4	2-1	2-2	2-3	2-4	3-1	3-4	3-2	3-3	4-2	4-1	4-3	4-4

Figure 2. Chromosome Genes and Machines

So the chromosome of machine 1 is 4 1 3 2, and the chromosome of machine 2 is 2 3 1 4, and the chromosome of machine 3 is 2 4 1 3, and the chromosome of machine 1 is 1 2 3 4.

B. Fitness and Selection

Fitness function is defined of each chromosome so as to determine which with reproduce and survive into the next generation. It is relevant to the objective function to be optimized. The greater the fitness of a chromosome is, the greater the probability to survive. In this study, the fitness function is defined as the function of the objectives function is expressed as $f = 1/u(f(x))$.

C. Crossover

Since sequencing as well as assignment problems allow a permutation encoding, various permutation crossover operators have been developed. The crossover process is used to breed a pair of children chromosome from a pair of parent chromosomes using a crossover method. A binary vector of equal length as the permutation is filled at random. This vector defines the order in which the operations are successively drawn from parent 1 and parent 2. We now consider the parent and offspring permutations and the binary vector as list.

(1) The crossover based on working procedure

Parent1 and Parent2 are selected as parent:

$$parent1 = (o_{11}, o_{13}, o_{12}, o_{21}, o_{22}, o_{24}, o_{31}, o_{32}, o_{33}, o_{34}, o_{42}, o_{43})$$
$$parent2 = (o_{14}, o_{12}, o_{13}, o_{21}, o_{23}, o_{22}, o_{31}, o_{33}, o_{32}, o_{41}, o_{43}, o_{44})$$

In this example, supposing that the processing order of job J_1 is o_{11}, o_{21}, o_{31}, and the processing order of job J_2 is $o_{12}, o_{22}, o_{32}, o_{42}$, and the processing order of job J_3 is o_{13}, o_{33}, o_{43}, and the processing order of job J_4 is o_{24}, o_{34}. The position of o_{11}, o_{21}, o_{31} are provided from Parent1, and the position of o_{12}, o_{22}, o_{32} is provided from Parent2. Then the Offspring1 position of o_{11}, o_{21}, o_{31} is drawn from parent1 and deleted from parent2, the other position are filled with parent2, so as to offspring2. Example of crossover is given as following:

$$offspring1 = (o_{11}, o_{14}, o_{12}, o_{21}, o_{13}, o_{23}, o_{31}, o_{22}, o_{33}, o_{32}, o_{41}, o_{43})$$
$$offspring2 = (o_{11}, o_{12}, o_{13}, o_{21}, o_{24}, o_{22}, o_{31}, o_{33}, o_{32}, o_{34}, o_{42}, o_{43})$$

(2) The crossover based on machine distribution

In here, we consider one-point crossover. A crossover point is selected from two parents randomly. Example of crossover is given in figure 3 and figure 4.

	Working sequence	O_{11}	O_{13}	O_{12}	O_{21}	O_{22}	O_{24}	O_{31}	O_{32}	O_{33}	O_{34}	O_{42}	O_{43}
parent1	machine	1	3	2	4	3	1	2	4	2	1	3	4

	Working sequence	O_{14}	O_{12}	O_{13}	O_{21}	O_{23}	O_{22}	O_{31}	O_{33}	O_{32}	O_{41}	O_{43}	O_{44}
parent2	machine	4	2	3	1	3	2	4	1	2	1	3	4

Figure 3. Distributed machine from parent

The position 7 is selected randomly. Two parents both have the operation, o_{12} , o_{13} , o_{21} , o_{23} , o_{31} before position 4, and permuting distributed machine. Then, Two parents have the operation o_{32} , o_{33} , o_{43} after position 7, and permuting distributed machine. Such operation we obtain the feasible solution. The result is as following:

	Working sequence	O_{11}	O_{13}	O_{12}	O_{21}	O_{22}	O_{24}	O_{31}	O_{32}	O_{33}	O_{34}	O_{42}	O_{43}
offspring1	machine	1	3	2	1	2	1	4	2	1	1	3	3

	Working sequence	O_{14}	O_{12}	O_{13}	O_{21}	O_{23}	O_{22}	O_{31}	O_{33}	O_{32}	O_{41}	O_{43}	O_{44}
Offspring2	machine	4	2	3	4	3	3	2	1	4	1	4	4

Figure 4. Distributed Machine from Offspring

D. Mutation

The mutation operation is critical to the success of the GA since it diversifies the search directions and avoids convergence to local optima. We select a parent, and an operation is get randomly. As an example that an operation O_{23} is between O_{11} and O_{31} due to the same operation having successive sequence and we insert it.

E. Genetic Algorithm Flow Chart

Step1 Initialization population is generated randomly, and it is feasible schedule.

Step2 The fitness is defined by objective of JSP model, and individual adaptive value is evaluated.

Step3 The crossover is operated in the population according to probability of crossover Pc, so the offspring is generated.

Step4 The individual is selected randomly according to probability of mutation P_m, so the offspring is generated.

Step5 The new individual adaptive value is calculated, parent and offspring are taken part in survival competition together.

Step6 Adjusting the termination criterion, then the optimal solution is obtained, otherwise going back to Step3.

IV. Simulation Study

In order to investigate the effectiveness of the proposed algorithm, the experiments were conducted based on the production data. The experiment were conducted under five machines and eight processes, and each process has P=[2,5,2,3,4,4,2,3] machines number. We consider initial population that is 50, and iteration is 50, and probability of mutation is 0.2. The process time is brought randomly. The result is as followed:

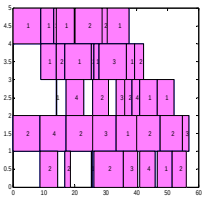

Figure 5. Gantt chart

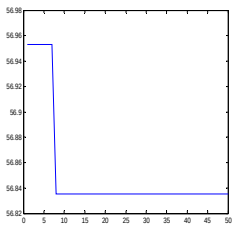

Figure 6. Minfitness convergence curve.

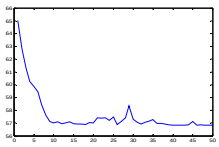

Figure 7. Minfitness convergence curve

TABLE I.
THE RESULT OF MINFITNESS AND MEANFITNESS

count	minfitness	meanfitness
1	56.9531	62.8344
2	56.9531	61.2899
3	56.9531	60.2423
4	56.9531	59.8793
5	56.9531	59.4404
6	56.9531	58.4496
7	56.8354	57.6321
8	56.8354	57.1184
9	56.8354	57.0184
10	56.8354	57.0974
11	56.8354	56.9453
12	56.8354	57.0105
13	56.8354	57.0835
14	56.8354	56.9374
15	56.8354	56.9257
16	56.8354	56.9178
17	56.8354	56.9060
18	56.8354	57.0425
19	56.8354	57.0069
20	56.8354	57.4327
21	56.8354	57.4010
22	56.8354	57.4064
23	56.8354	57.2132
24	56.8354	57.4837
25	56.8354	56.9046
26	56.8354	57.1388
27	56.8354	57.4319
28	56.8354	58.3622
29	56.8354	57.3023
30	56.8354	57.0707
31	56.8354	56.9322
32	56.8354	57.0661
33	56.8354	57.1411
34	56.8354	57.2817
35	56.8354	56.9783
36	56.8354	56.9783
37	56.8354	56.9210
38	56.8354	56.8612
39	56.8354	56.8354
40	56.8354	56.8354
41	56.8354	56.8354
42	56.8354	56.8354
43	56.8354	56.8354
44	56.8354	56.8354
45	56.8354	56.8354
46	56.8354	56.8354
47	56.8354	56.8354
48	56.8354	56.8354
49	56.8354	56.8354
50	56.8354	56.8354

TABLE II.
THE BEGINNING TIME OF EACH PROCESS ON EACH MACHINE

process / job	1	2	3	4	5	6	7	8
1	0	9.1012	13.2199	13.9644	19.8049	20.0597	28.7669	30.5516
2	9.1012	14.0056	16.7928	25.3366	26.1488	27.9002	36.6639	39.3902
3	14.0056	17.2294	25.8068	33.3904	36.1150	38.5455	41.0082	46.6215
4	0	8.7275	17.2294	25.8068	33.3904	40.0044	47.6074	54.8757
5	8.7275	16.7928	25.3366	26.1488	35.7055	41.0082	46.6215	51.5643

TABLE III.
THE END TIME OF EACH PROCESS ON EACH MACHINE

process / job	1	2	3	4	5	6	7	8
1	9.1012	13.2199	13.9644	19.8049	20.0597	28.7669	30.5516	37.4729
2	14.0056	16.7928	25.3366	26.1488	27.9002	36.6639	39.3902	42.2601
3	14.1233	22.8649	30.9535	36.1150	38.5455	41.0082	46.6215	52.0595
4	8.7275	17.2294	25.8068	33.3904	40.0044	47.6074	54.8757	56.8354
5	14.4420	18.5329	25.6951	35.7055	40.2717	45.9691	51.5643	56.0003

TABLE IV.
THE MACHINE NUMBER OF EACH PROCESS ON EACH JOB

process / job	1	2	3	4	5	6	7	8
1	1	1	1	1	2	2	2	1
2	1	2	1	2	1	3	1	2
3	1	4	2	3	2	4	1	1
4	2	4	2	3	1	2	2	3
5	2	2	1	2	3	4	1	2

The above example shows the makespan is 56.8354. The minfitness is attained 56.8354, when the iterative number is 7 in the figure 2. The meanfitness swings around 56.8354, when the iterative number is 39 that the meanfitness attains stabilization.

V. CONCLUSIONS

This paper proposed a GA for solving the agile job shop scheduling to minimize the makespan. Two-row chromosome structure is designed based on working procedure and machine distribution. The relevant crossover and mutation operation is also given. Finally, the Gantt chart is drawn based on five machines and eight processes. The computational result shows that GA can obtain better solution. The minfitness and meanfitness validate the solution that is validity.

REFERENCES

[1] Lars Monch, Rene Schabacker, Detlef Pabst et al,"*Genetic Algorithm-based Sub-problem Solution Procedures for a Modified Shifting Bottleneck Heuristic for Complex Job Shops"*, European Journal of Operational Research, Vol.3 No.177,pp.2100-2118，2007

[2] Young Su Yun, "*Genetic Algorithm with Fuzzy Logic Controller for Preemptive and non-Preemptive Job Shop Scheduling Problems"*, Computers & Industrial Engineering, Vol.3 No.43 pp.623-644，2007

[3] Dessouky, M.M, Leachman et al, "*Dynamic models of production with multiple operations and general processing times"*, Journal of the Operational Research Society, Vol.6 No.48 pp:983-997，1997

[4] Makoto Asano, Hiroshi Ohta, *"A Heuristic for Job Shop Scheduling to Minimize Total Weighted Tardiness"*, Computers & Industrial Engineering, Vol. 2-4 No 42 pp:137-147，2002

[5] Gomes, M.C., Barbosa-Povoa et al, *"Optimal scheduling for flexible job shop operations"*, International Journal of Production Research, Vol.11 No.43 pp:2323-2353，2005

[6] Taicir Loukil, Jacques Teghem, Philippe Fortemps , *"A multi-objective production scheduling case study solved by simulated annealing"*, European Journal of Operation Research, Vol.3 No.179 pp:709-722，2007

[7] Jason Chao-Hsien Pan, Jen-Shiang Chen, *"Mixed–Binary Integer Programming Formulations for the Reentrant Job Shop Scheduling Problem"*, Computers & Operations Research, Vol.5 No.32 pp:1197-1212，2005

[8] S.Q.Liu, H.L.Ong,K.M.Ng(2005), " *Metaheuristics for Minimizing the Makespan of the Dynamic Shop Scheduling Problem"*, Advances in Engineering Software, Vol.3 No.36 pp: 199-205

[9] Bortjan Murovec, Peter Suhel, *"A repairing technique for the local search of the job-shop problem"*, European Journal of Operational Research, Vol.1 No.153

pp:220-238，2004

[10] Hiroshi Ohta, Toshihiro Nakatani, *"A heuristic job-shop scheduling algorithm to minimize the total holding cost of completed and in-process products subject to no tardy jobs"*, International Journal Production Economics, Vol.1 No.101 pp:19-29，2006

[11] Chen Hua-ping, Gu Feng, Lu Bing-yuan et al, *"Application of Self-adaptive Multi-objective Genetic Algorithm in Flexible Job Shop Scheduling"*, Journal of System Simulation, Vol.8 No.18 pp: 2271-2274，2006

[12] Yu Yi-wei, Huang Tie-qun, Ye Liang-peng, *"Research on bi2objective flexible job shop scheduling based on genetic algorithm"* Journal of China Jiliang University, Vol.3 No.17 pp: 246-250，2006

[13] Hong Zhou, Yuncheng Feng, Limin Han, *"The Hybrid Heuristic Genetic Algorithm for Job Shop Scheduling"*, Computers & Industrial Engineering, Vol.3 No.40 pp:191-200，2001

[14] Byung Joo Park, Hyung Rim Choi, Hyun Soo Kim, *"A Hybrid Genetic Algorithm for the Job Shop Scheduling Problems"*, Computers & Industrial Engineering, Vol.4 No.45 pp: 597-613，2003

Ye Li and received PH.D degree from Dalian Maritime University in 2006. She current research interests include supply chain management, system modeling and production planning.

Yan Chen and received PH.D degree from Dalian University of Technology. She has been professor of Dalian Maritime University. Her main research interests include system modeling, management science and engineering et al.