A Knowledge-Based Genetic Algorithm for the Job Shop Scheduling Problem

HUNG-PIN CHIU
Department of Information Management
Nanhua University
32,Chung Keng Li, Dalin, Chia-Yi , 622 Taiwan, R.O.C.

KUN-LIN HSIEH
Department of Information Management
National Taitung University
684, sec.1, Chung-Hua Rd., Taitung, 950 Taiwan, R.O.C.

YI-TSUNG TANG
Department of Computer Science and Information Engineering
National Cheng Kung University
1, University Road, Tainan City 701, Taiwan, R. O. C.

WAN-JUNG CHIEN
Department of Information Management
Nanhua University
32,Chung Keng Li, Dalin, Chia-Yi , 622 Taiwan, R.O.C.

Abstract: - This study presents a novel use of attribution for the extraction of knowledge from job shop scheduling problem. Our algorithm improves the traditional GA and using knowledge to keep the quality of solution. Based on the knowledge, the search space will be leaded to a better search space. In addition, this study uses mutation to do local search and refresh the knowledge and population when the solution fall into local minimum. Based on those methods, our algorithm will have the intensification and diversification. Those can make the algorithm have good convergence and leap for the search space to find the better solution. The experiment results show that algorithm steadily and can find the approximate optimal solution. And the knowledge is useful in provide the gene selection information.

Key-Words: - hybrid GA, knowledge, Job-shop scheduling problem.

1 Introduction

Scheduling problems exist everywhere in real-world circumstance, especially in the flexible manufacturing world. Many people pay close attention to it because poor scheduling can lead to higher cost for manufacturers and consequently higher prices for customer. Therefore, if we want to have the better efficiency, we must have a good schedule to promote the efficiency and reduce the time in the manufacturing process. Nevertheless, scheduling problems are categorized into different groups in the different machine environment (e.g., single machine problems, parallel machine problems, job shop problems, etc,). In those groups, job shop problem (JSP) emphasizes the order of job in the every machine. In the other word, it considers the order of every operator of job but not prescribes which machine is the first machine for the job. As a result, JSP is more complicated than other scheduling problem. Therefore, this study has focused on the JSP problem.

JSP is among the hardest combinational optimization problem. Most of the researches used different approaches to solved JSP such as: Tabu search [6, 12], simulated annealing [15, 19], ant colony system [1], neural network algorithm [2],

genetic algorithm (GA) [5, 7, 13, 20], and others. GA-based approach was used to solve JSP problem considerably in recent years among these studying. Cheng, Gen, and Tsujimura [17] have given a detailed sort survey on papers using GA to solve classical JSP in Part Ⅰ survey. Nevertheless, using traditional GA can't give consideration convergence rate, quality of solution and stability of search process. On the other hand, this algorithm can't balance intensification and diversification. Ignore the intensification will spend much time to search. And disregard the diversification will fall into local minimum easily. So many researches tried modify GA with other algorithm. Cheng [18] discuss various hybrid GA to solve JSP.

In recent years, many researches wanted to improve intensification or diversification. Mattfeld and Bierwirth [3] used a heuristic reduction of search space which can help GA to find better solution in a shorter computation time. Goncalves, Mendes, and Resende [11] constructed the scheduling to generate parameterized active schedules and used a local search heuristic to improve the solution in evolutionary process of GA. To sum up, there studies focused on improving the search space. Therefore, better solutions could be expected but the quality of solutions could not be guarded. Watanabe, Ida and Gen [14] use GA with modified crossover operator for JSP problem. It made use of random number to decide what gene must be reserved for children chromosome. If the offspring do not conform to constrain the JSP problem, it will be regulate by some rules. This paper changed the traditional crossover operator and considered influence of each gene. Nevertheless, using the random number to decide which gene can be retained to offspring did not exclude random effect.

In order to keep the quality of solutions, some studies used the better chromosome to replace the bad chromosome. This method is accomplished by first coping some of the best individuals from each generation to the next, in what is called an elitist strategy [7]. Chang, Hsieh and Hsiao [16] reserved some better chromosomes and replaced some bad chromosomes in each generation. Those methods supposed that if there is a better population, it will the easy to produce the better offspring in crossover operator. However, it was not exactly so and it may easy to fall into local minimum. For this reason, we propose the idea that if we can evaluate the fitness for genes and choose the better gene to generate the offspring which may lead to a better solution. And if reserving the better chromosomes can help the quality of solution, those better chromosomes may be have useful information for finding the better solution.

Based on those ideas, we will collect some best solutions by GA to sort some knowledge and use it to evaluate the fitness for gene. And then make use of concluded result to design the suitable crossover operator for JSP problem. Hope to use this idea to speed up the convergence and improve the solution for JSP problem. Beside, we use mutation to do the local search, hope this can keep the diversification and avoid intensification overly. We will describe the design and the logic behind this method. And use the experiment to demonstrate the feasibility. This research is a new attempt and which can apply to other optimization problem. Therefore, it is a very important problem and merit discussion about it.

2 Literature review

2.1 Job-shop scheduling problem (JSP)

JSP problem has been described as follows [4]: there are m different jobs and n different machines to be scheduled. Each job is composed of a set of operation and the operation order on each machine is prespecified. The required machine and the fixed processing time characterize each operation. A schedule is an allocation of the operations to time intervals on the machines. According to the allocated operation sequences in a schedule, the time required to complete all jobs is called makespan of the schedule. Table 1 shows a 3×3 JSP problem and concluded operations, job number, machine number, process time. For example, when we observe J1 and O1, it means that operation 1 of job 1 be arranged for machine 2 (M2) and spend 2 time units.

Table 1. Example of 3×3 JSP problem

	Operations		
Job	**O1**	**O2**	**O3**
J1	M2(2)	M3(3)	M1(7)
J2	M3(1)	M1(9)	M2(3)
J3	M1(3)	M2(8)	M3(5)

2.2 Genetic Algorithm for Job-shop Scheduling Problem

Genetic algorithm (GA) is one of the stochastic search algorithms based on biological evolution. In order to solve a clearly defined problem and an offspring represented the candidate of solutions. GA is according to crossover and mutation operators with their probabilities to produce a set of offspring

chromosomes. As we know, GA likes an over and over process, an iteration is called a generation. A run means the whole set of generations. We try to find one or more highly fit chromosomes.

Recently, there have more and more papers used hybrid GA to solve optimum problem. Because of GA provides quite simple structure, process and it has strong abilities of solving and searching. Furthermore, GA searches multiple points in search space of population by evolution of generations and characteristic of search randomly. The abilities can avoid GA dropping in the local optimum and toward the global optimum. Whitley [3] introduced designing GA has two important issues: selection pressure and population diversity. Selection pressure leads GA to exploit information from the fitter individuals and produces more superior offspring iteratively. The diversity in GA is concerned about the population, which contains a certain number of encoded individuals for exploration. Therefore, we must to find a good tradeoff between exploration and exploitation consideration of both convergence speed and optimized solution quality. Masato etc. [10] proposed the modified GA with search area adaptation (mGSA) for solving JSP that does not need such crossover operator in GSA. Goncalves etc. [7] presented a hybrid genetic algorithm for the job-shop scheduling problem. It used the chromosome representation of the problem is based on random keys. The scheduled used a priority rule in which are defined by GA.

3 The proposed modified TGA

Our algorithm was modified GA's deficiency. We use some better solutions (chromosomes) to collect knowledge and designing a eugenic crossover. However, those better solutions just bring the limited information. We can understand the machine number and processing time for this operation, but we can not collect the integrate information. Therefore, we design the operation table to classify those operations before collecting knowledge. Use knowledge to decide the fitness of gene in crossover operator and adjust the chromosome.

Hsieh and Hsiao [16] reserved some better chromosomes and replace some bad chromosome in each generation and improve the solution of GA. According to this result, we can assume that those better chromosomes may be included some useful information for improve solution. But GA can not demonstrate repeat-ability or provide an explanation of how a solution is developed. For this reason, we can't induce information from the solutions of GA. Therefore, we will use the method which was brought up by Koonce and Tsai [4]. This method used attributions to induce information from the solution of GA.

Better solutions (chromosome) may be have some information and can help us to find optimal solution. Therefore, we could take advantage of those better solutions to collect knowledge. The KGA process was described as following:

Step1: Create initial population.

We used random number to produce some chromosomes.

Step2: Compute fitness value.

Transform the makespan into the fitness value. If the chromosome has lesser makespan, it will have the higher fitness value.

Step3: Generate new generation.

When we have initial population and fitness value, we will use those data to do the next step.

Step4: Select population.

We use the roulette wheel method to select two chromosomes. This step is the same with GA.

Step5: Crossover.

In the blended crossover, we must select which crossover operator by generation number. If the child was not better than one of parents, we must do the mutation.

Step6: Mutate.

In the forced mutation, we will mutate the child which is not better than parents in crossover.

Step7: Meet the population size or not.

If this generation has enough population, this generation will be over.

Step8: Compute fitness value.

In this process, we will compute fitness value of new population and using new population in the next generation.

Step9: Reach the generation number or not.

If the KGA process has enough generation number, the algorithm will be over. Otherwise, continuing the KGA and determine the solution fall into local minimum or not.

Step10: If the solution fall into local minimum or not.

If the best solution in each generation had not change several times, we will determine the solution has fall into local minimum. When the solution does not fall into local minimum, we will collect the better solution from this generation. The collecting of the better solution is the same the part 1. If the solution is better than one of the better solution which was retained, we will retain this solution until the solution fall into local minimum. When the solution was fells into local minimum, we will collect new knowledge and refresh knowledge by new knowledge.

Step11: Refresh population

When the solution fell into local minimum, we must refresh knowledge. And this situation was represented this search space that can't find the better solution. So we must refresh the population and search other space again. In the refresh population process, we will sort the original population by fitness value and selecting the first 20% population to new population. The other new population was produced by random number.

4 Experiential results

The following experiments showed the 10×10 JSP benchmarking problems solving only for the purpose of illustrating the computational procedure discussed above. In this experiment, we use KGA for the 10×10 benchmark problem. This problem was generated by Fisher and Thompson. Lawler et al. [6] report that within 6000s when applying a deterministic local search to this problem and find more than 9000 local optima. It is perceived that this problem has the difficult to find the optimal solution. Besides, it was proved that the optimal makespan is 930. We can use this result to determine whether the solution by KGA is good or not. Figure 1 shows the result by GA and KGA. In this Figure 1, the best solution by KGA is 936 and by GA is 1053. And it just spent 440 generations to find the makespan 964. This result proved that KGA had faster convergence than GA and its result better than GA. Table 2 shows the progress of the 10×10 benchmark instance. According to this table, we can know that we did not find the optimal makespan, but the solution by our algorithm is very close the optimal makespan. Figure 2 shows the makespan for KGA for 100 time trial. We can find that most of the solutions fall into the range between 960 and 969. This result can represent our algorithm is steady. And the best solution by KGA is 936. This solution is not the optimum solution, but it is close the optimum solution. For those result, we can prove KGA is useful and using the knowledge can improve the quality of solution.

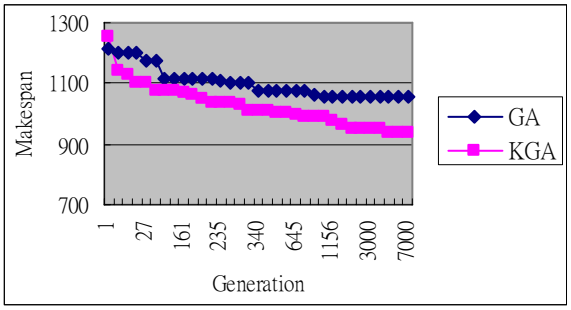

Figure 1. The makespan of the 10×10 benchmark problem

Table 2. Progress of the 10×10 benchmark instance

Researchers who achieved solution	Makespan	Researchers who achieved solution	Makespan
Fisher and Thompson(1963)	1101	Lageweg(1982)	935
Balas(1969)	1177	Fisher et al.(1983)	1084
Schrage(1970)	1156	Lageweg(1984)	930
Florian et al.(1971)	1041	Barker and McMahon(1985)	960
Bratley et al.(1973)	980	Adams et al.(1988)(sbII)	930
McMahon and Florian(1975)	972	Carlier and Pinson(1989)	930
Lageweg et al.(1977)	1082		

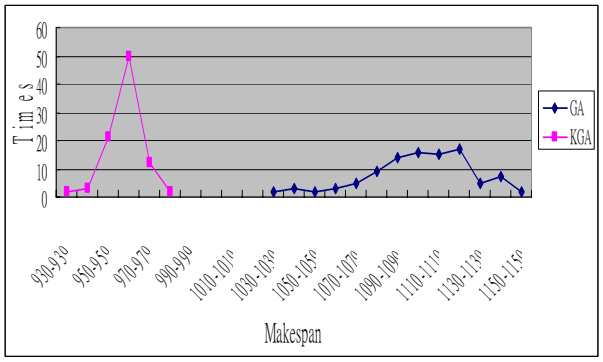

Figure 2. The makespan for KGA (run 100 times)

5 Conclusion Remarks

According to those experiments, we can obtain some conclusions. The first, the knowledge is useful. In the eugenic crossover, the knowledge was used to evaluate the fitness-gene and retained the higher fitness-gene in offspring. This method can raise the quality of offspring and produce better offspring. Based on this result, the knowledge is useful in provide the gene selection information. But this method will make the algorithm fall into local minimum easily. This is because when we can find the better solution than GA, the knowledge becomes useless. Therefore, the knowledge must be renewed in KGA process. The second, this algorithm can raise the convergence. Because the algorithm used the knowledge to improve the crossover, it will be leaded to search the specific space. For this reason, the method could search out the better offspring in short time and raised the convergence. The third, the algorithm can balance the intensification and diversification. This algorithm used the knowledge to search special space and improve the convergence. Therefore, this method achieves the intensification which makes the algorithm to search the space where better solution exists.

References:

[1] A. Colorni, M. Dorigo, V. Maniezzo and M. Trubian, “Ant system for Job-shop Scheduling”, Statistics and Computer Science, vol. 34, pp.39-53, 1994.

[2] A. S. Jain and S. Meeran, “Job-Shop Scheduling Using Neural Networks”, International Journal of Production Research, vol.36, No.5, pp.1249-1272, 1998.

[3] C. Dirk Mattfeld and Christian Bierwirth, “An efficient genetic algorithm for job shop scheduling with tardiness objectives”, European journal of operational research, vol. 155, pp.616-630, 2004.

[4] D.A. Koonce and S.-C. Tsai, “Using data mining to find patterns in genetic algorithm solutions to a job chop schedule”, Computers & Industrial engineering, vol.38, pp.361-374, 2000.

[5] E. Falkenauer, S. Bouffoix, “A genetic algorithm for job shop”, Proc. of the 1991 IEEE international Conference on Robotics and Automution, 1991.

[6] E. Nowicki, C. Smutnicki, “A Fast Taboo Search Algorithm: for the Job Shop Problem”, Managemenr Science, vol. 42, pp. 797-813, 1996.

[7] D.E. Goldberg, “Genetic Algorithms in Search”, Optimization, and Machine Learning, Addison-Wesley, USA, 1989.

[8] E.L. Lawler, J.K. Lenstra and Rinnooy Kan etc., ”Sequencing and scheduling: Algorithms and complexity”, Hardbook in operations research and management, 1993.

[9] G. Syswerda, “A study of reproduction in generational and steady-state genetic algorithms”, foundations of Genetic Algorithms, pp. 94-101, 1991.

[10]J. Carlier and E. Pinson, “An algorithm for solving the job shop problem”, Management science, vol. 35, pp.164-176, 1989

[11]J. F. Goncalves, M. Mendes, and Maurício G.C. resende, “A hybrid genetic algorithm for the job shop scheduling problem”, European journal of operational research, vol. 167, pp.77-95, 2005.

[12]K. Morikawa, T. Furuhashi, Y. Uchikawa., “Single Populated Genetic Algorithm and its Application to Job-shop Scheduling”, Proc. of Industrial Electronics, Control, Instrumentation, and Automation on Power Electronics and Motion Control, pp. 1014-1019, 1992.

[13]L. Davis, “Applying adaptive algorithms to epistatic domains”, In Proc. of the Inter. Joint Con5 on Artificial Intelligence, pp. 162-164, 1985.

[14]M. Watanbe, K. Ida, and M. Gen, “A genetic algorithm with modified crossover operator and search area adaptation for the job-shop scheduling problem”, Computers & Industrial Engineering, vol. 48, pp.743-752, 2005.

[15]M. Kolonko. Some new results on simulated, “annealing applied to the job shop scheduling problem”, European Journal of Operational Research, pp. 123-.136, 1999.

[16]P.C. Chang, J.C. Hsieh and C.H. Hsiao, “Application of genetic algorithm to the unrelated parallel machine scheduling problem”, Chinese industrial of Industrial Engineers, vol. 19, pp.79-95, 2002.

[17]R. Cheng, M. Gen, and Y. Tsujimura. “A tutorial survey of job-shop scheduling problems using genetic algorithms---I:Representation”. Computers ind. Engng vol. 30, No. 4, pp.983-997, 1996

[18]R. Cheng, M. Gen, and Y. Tsujimura. “A tutorial survey of job-shop scheduling problems using genetic algorithms--- II: Hybrid genetic search strategies”. Computers & Engineering, vol. 36, pp.343-364, 1999

[19]V.L. PJM, A. EHL, and L. JK, “Job shop scheduling by simulated annealing”, Operations Research, 40, pp.113-125, 1992.

[20]X. Li, W. Liu, S. Ren, and X. Wang, “A solution of job-shop scheduling .problems based on genetic algorithms”, IEEE Intemational Conference on Systems, Man, and Cybernetics, vol. 3, pp. 1823 -1828, 2001.