A Tabu Genetic algorithm with Search Area Adaptation 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.

CHING-YU WANG
Department of Information Management
Nanhua University
32,Chung Keng Li, Dalin, Chia-Yi , 622 Taiwan
R.O.C.

Abstract: - The job-shop scheduling problem is the important issues to the research of optimal problems. Besides, tabu search is also applied to GA, called TGA for traveling salesman problem (TSP) that has better effectiveness than GA. Thus, this is an interest and important research area for job-shop scheduling problem with TGA. In this paper, we try to discuss this issue. According to the TGA, it maintains diversity through broad-sense incest prevention. Therefore, the solutions can contain theirs diversity and prevent premature convergence. But in JSP problem, the crossover and mutation manners of TGA cannot produce the better solutions than GA. So we modified the crossover and mutation phases of TGA called modified TGA (MTGA). First, the modified crossover search phase uses a threshold (TH_c) to control the times of crossover for improving the qualities and convergence of solutions. Second, the mutation search phase use two parameters to control the selected points and the times of mutation in order to make the global search wildly and prevent to drop into local minimum more easily. And the experiments results demonstrate the superiority of MTGA in job-shop scheduling problems. Not only balance intensification, but also diversification.

Key-Words: - Genetic algorithm, Job-shop scheduling problem, Tabu search

1 Introduction

Current market trends, shorter product life cycles and competitive pressure to reduce costs have resulted in the need for zero inventory systems. In order to maintain market share, the system must be fast responding which implies that more stock has to be maintained. These conflicting requirements demand efficient, effective and accurate scheduling which is complex in all but the simplest production environment [1]. So, scheduling problems need to be solved by good scheduling algorithms and heuristics. Job-shop scheduling problem (abbreviated to JSP) is one of the hardest well-known combinatorial optimization problems. The problem is an allocation of the operations to time intervals on the machines, in order to find a minimum makespan.

Genetic algorithms (GAs) are well-known heuristic algorithms and have been applied to solve a variety of complicated problem. In recent years, there is an interest of in using genetic algorithms for solving JSP [7, 8, 10]. GA has shown a good performance regarding its ability to search globally. It searches multiple points in the search space of population. It also uses a crossover operator that enables to search wider region. The diversity in GA is attributed to the form of population, which contains a certain number of encoded individuals for population. Therefore, heuristic algorithms pursue a good balance between exploration and exploitation in consideration of both convergence speed and optimized solution quality. However, in the conventional GA, parents are approved without any further examination after they are chosen at random or just by fitness. So, it's hard to prevent the ancestry mating and control the solution quality. In this research, we adapt tabu genetic algorithm and to modify tabu genetic algorithm, called MTGA that with local search mechanisms in crossover and mutation search phases to contain the diversification and intensification of solutions. Finally, using several experimental results to prove MTGA has better performance than GA and TGA.

2 Literature review

2.1 Job-shop scheduling problem (JSP)

In general, the classical JSP can be stated as follows [6]: There are *n* different jobs and *m* different machines to be scheduled. Each job is composed of a set of operations and the operations order on machines is prespecified. The required machine and the fixed processing time characterize each operation. The problem is to determine the operation sequences on the machines for minimizing the makespan, and the time is necessary to complete all jobs. An example of the three-job three-machine (3×3) JSP problem is presented in Table 1.

A chromosome has gene information that shows the order of the job-number for solving the problem in GA. If the number of jobs is n and the number of machines is m, the chromosome consists of $n \times m$ genes. Each job must be depending on an order relation; it will appear m times exactly. Thus, each chromosome represents a feasible solution. In a 3×3 job-shop scheduling problem, 9 genes denote a chromosome, '1' shows job1 (J_1), '2' shows job2 (J_2), '3' shows job3 (J_3), respectively. Each gene, operation (*Op*), is given priority from left to right. Therefore, a left gene has higher priority than right one. An example of a chromosome in a 3×3 job-shop scheduling problem is shown in Figure 1.

Table 1. Example of 3×3 JSP problem

	Operation (machine number /processing time)		
Job	Op_1	Op_2	Op_3
J_1	1 / 3	2 / 3	3 / 2
J_2	1 / 1	3 / 5	2 / 3
J_3	2 / 3	1 / 2	3 / 3

Chromosome	2	1	3	3	2	1	3	2	1

Figure 1. A Chromosome of 3×3 JSP problem

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.

2.3 Tabu Genetic Algorithm (TGA)

Fred Glover (1989) [5] proposed TS that is a strategy for solving combinatorial optimization problems. In this section, we display TS in a simple form of its conceptions. TS constrains the search by classifying certain of its moves as forbidden and to free the search by a short-term memory function. And utilize aspiration criteria to override the tabu restriction that allow superior solution. In TS, the tabu restrictions and aspiration criteria played a dual role in constraining and guiding the search process. Besides, it uses memory function to record the moving trajectories. According to the used memory structure, we classify these approached into two major categories: computation-based and memory-based mating strategies [2].

Ting, Li and Lee proposed tabu genetic algorithm [2], which by incorporates the feature of TS into GA's selection. TGA integrates the tabu list to prevent inbreeding so that population diversity can be maintained, and adapts the aspiration criterion to provide moderate selection pressure. Then it utilizes the self-adaptive mutation to overcome the hard of deciding mutation rate. It uses the classic traveling salesman problem (TSP) as a benchmark to verify the effectiveness of the proposed algorithm. If the part of TS, highlighted in gray, is ignored, TGA becomes into a simple GA. It means that the reproductions of GA needed to be supervised by the elements of TS. After being produced by genetic operators, each pair of offspring went through these restricts of TS, tabu list and aspiration criterion, to confirm that their parents are allowed to mate. When the mating is allowed to mate or is good enough to meet the aspiration criterion, it is grouped into acceptable and the offspring are reserved for the subpopulation. Otherwise, the offspring are rejected and the process returns to select a mate. The process is repeated until the mating is acceptable. If the number of experiments exceeds a predefined threshold, this condition is thought as a Deadlock. Then the selected mate will be mutated. They are regarded as acceptable mate, and delivered to subpopulation. All the researches have indicated that genetic algorithm and job-shop scheduling problem are the important issues to the research of optimal problems. Besides, tabu search is also applied to GA, called TGA for traveling salesman problem (TSP) that has better effectiveness than GA [2]. Thus, this is an interest and important research area for job-shop scheduling problem with TGA.

3 The proposed modified TGA

In TGA for JSP, the results and effectiveness of experiments is not as good as the algorithm for TSP. It inherits the property of diversity, but it can't take the balance between diversification and intensification. Thus, we modified crossover and mutation search phases of TGA, called modified TGA (MTGA) to improve ability of local search and extend search area. And try to convergent solutions under the diversity. Based on the above definitions and discussion, the modified algorithm MTGA is formulated as follows. Assume that the population P consists of N chromosomes $C_1,\ldots, C_N$, with fitness $F_1,\ldots, F_N$, respectively. The best fitness S is recoded in each generation t. The genetic operators are performed to produce offspring $C'_1,\ldots, C'_N$, and the deadlock criterion TH is defined to prevent infinite loop. The algorithm terminates at t_{max} generations, at which point the obtained best fitness S is the optimized result. The algorithm of MTGA is showed in Figure 2.

MTGA1. [Initialization.] Set $t \leftarrow 0$, and initialize population P_t.
MTGA2. [Evaluation.] Evaluate population P_t and result in $F_1,\ldots, F_N$. Set $S \leftarrow \max(F_1,\ldots, F_N)$.
MTGA3. [New generation.] Set $n \leftarrow 0$ and $r \leftarrow 0$. (Where n is the number of produced offspring, and r is the repeated times of deadlock.)
MTGA4. [Select Parent A.] Set $i \leftarrow$ the best $(1,\ldots, N)$. (Select parents C_i.)
MTGA5. [Select Parent B.] Set $j \leftarrow$ random $(1,\ldots, N)$. (Select parents C_j.)
MTGA6. [Tabu?] If Tabu (C_i, C_j) = false, go to step MTGA10. (Check parents C_i and C_j to see if they are forbidden to mate.)
MTGA7. [Crossover.] $(C'_i, C'_j) \leftarrow$ Crossover (C_i, C_j).
MTGA8. [Aspiration?] If Aspiration (C'_i, C'_j) = true, go to step MTGA12.
MTGA9. [Deadlock?] Set $r \leftarrow r + 1$. If $r <$ TH, go to step MTGA5, otherwise, go to step MTGA11.
MTGA10. [Crossover processing.] Set count_1 $\leftarrow 0$, and TH_c is the repeated times of crossover.
(a) Select crossover points randomly.
(b) $(C'_i, C'_j) \leftarrow$ Crossover (C_i, C_j).
(c) If $(F_i,\ldots, F_j) < (F'_i, F'_j)$, go to step MTGA12.
(d.) Set count_1 $\leftarrow$ count_1 + 1. If count_1 $< TH_c$, go to step (a.). Otherwise, go to step MTGA12.
MTGA11. [Mutation Processing.] Set count_2 $\leftarrow 0$, and TH_m, is the repeated times of mutation.
(1) Select mutation points randomly.
(2) $(C'_i, C'_j) \leftarrow$ Mutation (C_i, C_j).
(3) Set count_2 $\leftarrow$ count_2 + 1. If count_2 $< TH_m$, go to step (1). Otherwise, go to step MTGA12.
MYGA12. [Subpopulation filled?] Insert the offspring C_i and C_j into P_{t+1}, and set $n \leftarrow n + 2$. If $n < N$, set $r \leftarrow 0$ and return to step MTGA4.
MTGA13. [Evaluation.] Set $P_{t+1} \leftarrow$ Survive (P_t, P_{t+1}) and $t \leftarrow t + 1$. Evaluation population P_t, and set $S \leftarrow \max(F_1,\ldots, F_N)$.
MTGA14. [Termination?] If $t < t_{max}$, return to step MTGA3. Otherwise, the algorithm MTGA is complete.

Figure 2. The algorithm of MTGA

3.1 Modified crossover search phase

In TGA, the crossover phase has to obey the tabu restrictions, so it maintained the diversity of population. However, it will due to the insufficient convergence, for this reason, the modified crossover search phase uses a threshold (THc) to control the times of crossover. If the children are better than the parents, they survived in subpopulation. Otherwise, the parents repeat to crossover until the times of crossover equal to the threshold, and keep the last offspring to the subpopulation. About diversity issue, TGA had good performance in it. But in TGA crossover phase, it's crossover one time, then delivery the offspring in subpopulation. It's a consequence on the lack of convergence. Therefore, we utilize the times of crossover to do some local search for improving the solutions qualities and converging faster.

3.2 Modified mutation search phase

Mutation operator is in order to adapt evolution environment and adjust it dynamically. But the chromosomes mutate by two selected points that are still too hard to outleap the local minimum. Hence, we added a mechanism in mutation processing. For example, assume a parameter set as 10, a pair from 1 to 10 be selected the mutation points randomly. This manner is in the course of making the global search wildly. It's also preventing to drop into local minimum more easily. After mutation, next step is another mechanism that controls the times of mutation by a threshold (TH_m). Finally, the best chromosomes are sent to the subpopulation.

4 Performance evaluation

In our research, several comprehensive experiments are conducted to evaluate the performance of TGA and MTGA in JSP. We apply ft 10, 10×10, benchmark problem which has received the greatest analysis is the instance generated by Fisher and Thompson [4]. Lawler et al [9]. report that within 6000 s when applying a deterministic local search to ft 10 more than 9000 local optima have been generated with a best makespan value of 1006, furthermore emphasizing the difficultly and hardness of this problem. Besides, several researches proved that the optimal makespan is 930 of ft 10.

4.1 Tabu list

The impact of the size of tabu list on the performance of TGA is considered. The parameter of proportionality, δ, determining the size of tabu list, is experimentally changed from 0.1 to 0.5. A population size (N = 50), (generation= 500) and deadlock (TH= 10) are simulated. Each parameter runs 20 times and then takes the best solutions for experiment result. The convergence of each parameter can be showed in Figure 3. By the number of parameter increased, the convergence is decreasing. When the parameters are 0.1 to 0.3, the convergences are still having the suspicion of premature convergence. But when the parameters are 0.4 and 0.5, the convergences are slower and slower. In these conditions can observe that the utility of tabu list can prevent premature convergence.

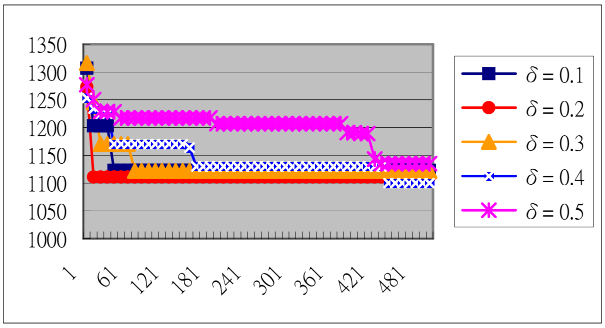

Figure 3. Comparison of tabu list parameter in TGA (10×10)

4.2 Performance comparison

In this experiment, we apply GA, TGA, TGA* (Selected the best chromosome as one of the parents, the other one selected randomly.) and MTGA to ft 10 JSP problem as a case. In TGA, TGA* and MTGA, the population size (N) is set to 100, the tabu list parameter is set to 0.4 and the deadlock is set to 20. In GA, the population size is set to 100, crossover rate = 0.5, and mutation rate = 0.15.

The performance experimental results are decomposed into two parts: MTGA is compared with GA, TGA and TGA*. And compared the frequency distribution of GA, TGA, TGA* and MTGA. In Figure 4, we can see that the performance of GA outperforms TGA. Although, TGA prevents premature convergence, but the qualities of solutions are not good enough. It's because that parents are selected by random. It leads the solution easier drop in local minimum. Thus, in TGA*, one of the parents choose the best chromosome in the population, the other one is selected by random. The qualities of solutions can be proved a lot. However, the performance of MTGA outperforms TGA*, the reason is that MTGA adapted location search in crossover and mutation search phases. It contains the good solutions in TGA* and also do local search in the good area that made the performance better. So the efficiency is obvious to the modified operator of crossover and mutation of MTGA and also found the optimal solution 930.

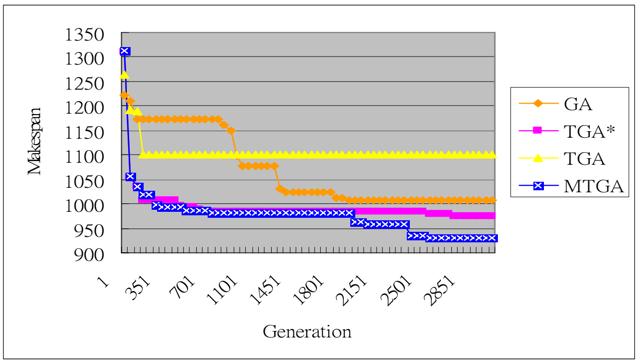

Figure 4. Comparison GA, TGA, TGA*and MTGA (10×10)

Figure 5 shows the frequency distribution of results, respectively. It is shown here that the distribution of MTGA has width narrower and the solutions are better than GA, TGA and TGA*. Therefore, it is confirmed that MTGA shows better performance than GA and TGA.

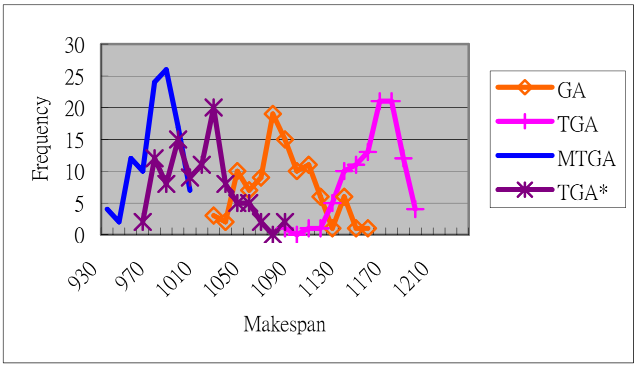

Figure 5. Frequency distribution of makespan (10×10)

5 Conclusion Remarks

This research apply TGA to 10×10 JSP problem, it maintains diversity through broad-sense incest prevention by tabu list. Therefore, the solutions can contain theirs diversity. It also can prevent premature convergence by deadlock. But in JSP problem, the crossover and mutation manners of TGA cannot produce the better solutions than GA. So we modified the crossover and mutation phases of TGA called modified TGA (MTGA). First, the modified crossover search phase uses a threshold (TH_c) to control the times of crossover for improving the qualities and convergence of solutions. Second, the mutation search phase use two parameters to control the selected points and the times of mutation in order to make the global search wildly and prevent to drop into local minimum more easily. And the experiments results demonstrate the superiority of MTGA in job-shop scheduling problems. Not only balance intensification, but also diversification. Furthermore, it finds out the optimal makespan 930.

References:

[1]A. S. Jain and S. Meeran. Deterministic job-shop scheduling: Past, present and future. European Journal of Operational Research 113, 1999, 390-434.

[2]Chuan-Kang Ting, Sheng-Tun Li, Chungnan Lee. On the harmonious mating strategy through tabu search, Information Sciences 156, 2003, 189-214, .

[3]D. Whitley, The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best, in: Proceedings of 3rd International Conference on Genetic Algorithms, San Mateo, CA, 1989, pp. 116–121.

[4]Fisher, H., Thompson, G.L., Probabilistic learning combinations of local job-shop scheduling rules. In: Muth, J.F., Thompson, G.L. (Eds.), Industrial Scheduling. Prentice-Hall, Englewood Cliffs, NJ, 1963, pp. 225-251.

[5]Fred Glover. Tabu Search — Part I. Operation Research Society of America, 1989.

[6]Gen, M., & Cheng, R.. Genetic algorithms and engineering design. New York: Wiley, 1997.

[7]Goncalves, Mendes and Resende. A hybrid genetic algorithm for the job shop scheduling problem. European Journal of Operational Reseach 167, 2005, 77-95.

[8]L. Davis. Job shop scheduling with genetic algorithm. In Porc. of the First Int. Conf. on Genetic Algorithms (Edited by J. Grefenstette), pp. 136-140. Lawrence Erlbaum Associates, Hillsdale, NJ, 1985.

[9]Lawler, E.L., Lenstra, J.K., Rinnooy Kan, A.H.G., Shmoys, D.B., Sequencing and scheduleing: Algorithms and complexity. In: Handbook in operations Research and Management Science 4: Logistics of Production and Inventory, 1993.

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