2011 3rd International Conference on Information and Financial Engineering
IPEDR vol.12 (2011) © (2011) IACSIT Press, Singapore

Artificial Chromosomes Embedded in Sub-population Gemetic Algorithm for a Multi-objective Scheduling Problems

Wang, Yen-Wen [1+], Wu, Jen-Long [2] , and Lin, Jong-Li [3]

[1] Department of Industrial Engineering and Management, Ching-Yun University, Taiwan.

[2] Ph.D. student of Department of Information Management, Yuan-Ze University, Taiwan.

[3] Department of Industrial Engineering and Management, Ching-Yun University, Taiwan.

Abstract. Sub-population Genetic Algorithms is a population-based approach for heuristic search in multiple objectives optimization problems. Different from the single objective problem, sub-population genetic algorithms is used to find the Pareto solutions of different objectives. However, the traditional mechanic in the genetic algorithms will diminish the searching space while evolving; it will cause the solutions converging too fast and fall into the local optima. In this research, two different kind of artificial chromosome operators will be introduced when the algorithm evolves to certain iteration for injecting to individual to search better combination of chromosomes, this mechanism will provide a more expansive searching space while evolving. The experiments result shows that these two operators possess fast convergence and average scatter of Pareto solutions simultaneously for solving multi-objective scheduling problems in test instances.

Keywords: Flowshop scheduling problem, Multi-objective scheduling, Artificial chromosome

1. Introduction

In the operations research literature, Flowshop scheduling is one of the most well-known problems in the area of scheduling. Flowshops are useful tools in modeling manufacturing processes. A permutation Flowshop is a job processing facility which consists of several machines and jobs to be processed on the machines. In a permutation Flowshop all jobs follow the same machine or processing order and job processing is not interrupted once started. Our objective is to find a sequence for the jobs so that the makespan and the completion time is a minimum.

The traditional mechanic in the genetic algorithms will diminish the searching space while evolving; it will cause the solutions converging too fast and fall into the local optima. Therefore, we take a close look at the evolutionary process for a permutation Flowshop scheduling problems and come out with the new idea of generating artificial chromosomes(ACs) to further improve the solution quality of the genetic algorithm. To generate ACs, it depends on the probability of each job at a certain position. The idea is originated from Chang et al.(2005) which propose a methodology to improve Genetic Algorithms (GAs) by mining gene structures within a set of elite chromosomes generated in previous generations. Instead of replacing the crossover operator and mutation operator due to efficiency concern, the probability model acquired from the elite chromosomes will be integrated with the genetic operators in generating artificial chromosomes, i.e., off-springs which can be applied to enhance the efficiency of the proposed algorithm. Apart from our previous researches, Wang(2010), Harik (1999), Rastegar (2006), Zhang (2005) have discussed and proved the genetic algorithm which is based on the probability models. There are only few researches in applying evolutionary algorithm with probability models to resolve discrete problems.

[+] Tel.: +88634581196; fax: +88634285319.
ywwang@cyu.edu.tw.

The rest of the research is organized as follows: Section 2 introduces the methodology of SPMA. In Section 3, extensive experiments are conducted to test the performance of the proposed algorithm in multi-objective Flowshop scheduling problems. Finally, the conclusion is discussed and future researches are also provided.

2. Problem statement

Flowshops are useful tools in modeling manufacturing processes. A permutation Flowshop is a job processing facility, which consists of several machines and several jobs to be processed on the machines. In a permutation Flowshop all jobs follow the same machine or processing order. Our objectives are to find a set of compromise solutions so that the makespan and maximum tardiness are minimized.

The Flowshop scheduling problem is a typical assembly line problem where n different jobs have to be processed on m different machines. All jobs are processed on all the machines in the same order. The processing times of the jobs on machines are fixed irrespective of the order in which the processing is done. The problem is characterized by a matrix $P = (p_{ij}), i = 1 \cdots n, j = 1 \cdots m$, of processing times. Each machine processes exactly one job at a time and each job is processed on exactly one machine at a time. The problem then is to find a sequence of jobs such that the makespan that is the completion time of the last job in the sequence on the last machine is minimized. If C_i denotes the completion time for job i, then we are trying to minimize max C_i. There are many other criterions that can be considered for optimization. We refer the reader to Bagchi for a detailed discussion of multi-objective scheduling using GA. For details of the Flowshop and other scheduling and sequencing problems we refer the reader to Baker(1974).

A more general Flowshop scheduling problem can be defined by allowing the permutation of jobs to be different on each machine. However, what work has been done to show on the more general Flowshop scheduling problem has tended to small improvement in solution quality over the permutation Flowshop scheduling problems (PFSP) while increasing the complexity of the problem substantially. The size of the solution space increases from $n!$ to $(n!)^m$. Other objective functions for the PFSP also received a lot of attentions. For example, the mean flow-time (the time a job spends in process), or the mean tardiness (assuming some deadline for each job) are to be minimized. Other real problems from the manufacturing industries such as jobs may have non-identical release dates, there may be sequence-dependent setup times, and there may be limited buffer storage between machines and so on. These characteristics of the real world problems will make the problem more complicated to be solved within a reasonable time frame. However, GA approaches provide a more realistic view to the problem. Since it can generate alternatives of sequences (in the evolving process each chromosome representing a feasible solution to the problem) to the decision maker, a more applicable sequence can be decided to solve the current problem with satisfactory results.

3. Methodology

Sub-population Genetic Algorithms(SPGA) with two ACs is proposed to solve Flowshop scheduling problems. Except for the original mining gene structures (Chang 2005), we have constructed a dominance matrix in the past research(Wang, 2010), we called AC2. However, we could found the traditional mechanic in the GAs will diminish the searching space while evolving; it will cause the solutions converging too fast and fall into the local optima. Thus, we have revised some mechanics in the AC2 and proposed AC3 in this research. Following shows the main task of AC3.

The detailed steps are described in the following:

To convert gene information into priority matrix:

In the original AC2, it records the relationship between each job and sequence as a dominance matrix, and injects the ACs by mining this matrix. We could observe the dominance matrix will cause the solutions fall into a local solution, especially in the multi-objective problem. In this research, we apply a priority matrix to replace the dominance matrix. In the priority matrix, if job i exists before job j, the frequency in the matrix is added by 1. To demonstrate the working theory of the AC generation procedure, a 5-job problem is illustrated. Suppose there are n chromosomes whose fitness is better than average fitness. Then, we accumulate the gene information from these n chromosomes to form a priority matrix. As shown in Figure 1.

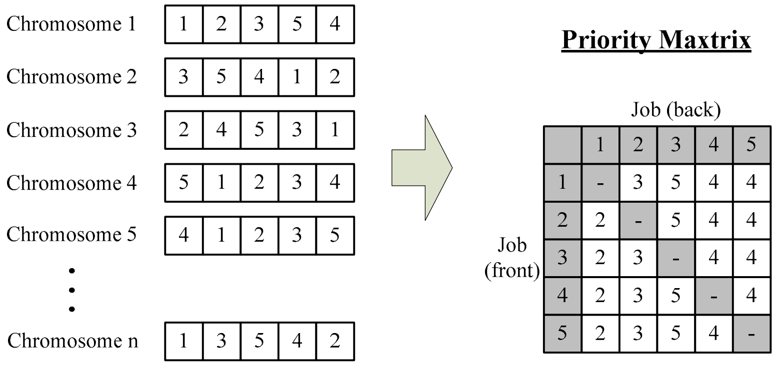

Fig 1. To collect gene information and converted into a priority matrix.

3.1. Generate artificial chromosomes:

As soon as we collect gene information into priority matrix, we are going to assign jobs onto the positions of each artificial chromosome. The first job is randomly assigned, and we select one job assigned to the following by roulette wheel selection (RWS) method based on the probability of each job was appeared after the front job. Then, the procedure continues to select the next job until all jobs are assigned. Assume the first job is to be assigned as job 2, which is shown in Figure 2. The frequency of following job is 8, 15, 7, and 5, the corresponding probability for job 1 is 8/35; job 3 is 15/35, and so on. Then, we apply roulette wheel select(RWS) to assigned the next job.

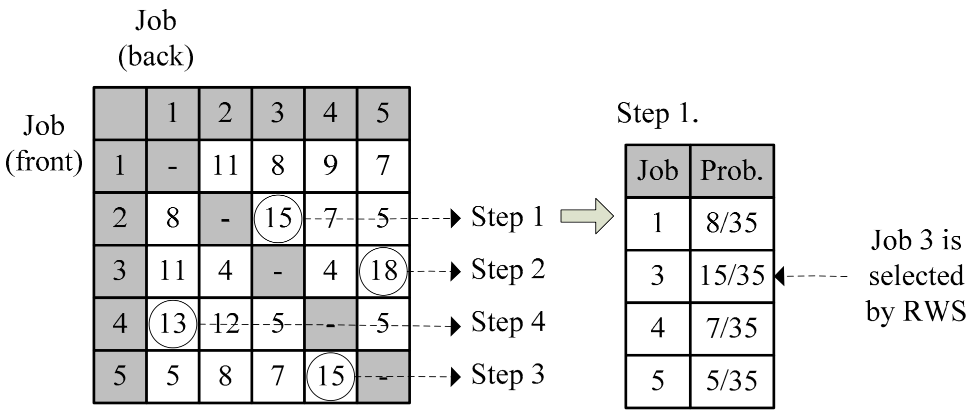

Fig 2. The probability of next job for job 2.

4. Experimental tests

The research uses the Flowshop scheduling case study by a standard benchmark (Reeves, 1995) with 30 jobs in 10 machines. Two objectives are the total completion time (Cmax) and maximum tardiness (Tmax). The experimental results will be compared with those of simple SPGA and Elite SPGA. The testing result of this sample problem is depicted in Table 1, Figure 3 and Figure 4.

TABLE I. THE EXPERIMENTAL RESULTS OF SPGA WITH DIFFERENT MECHANICS

Goal	Simple SPGA (SGA)	Elite SPGA (EGA)	SPGA-AC2 (AC2)	SPGA-AC3 (AC3)
Minimum solution for the generation	2301.83	2136.17	2142.40	2137.97
The STDEV of minimum solution for the generation	1.69	2.76	3.20	2.83
The STDEV of average solution for the generation	1.40	4.58	16.53	16.48

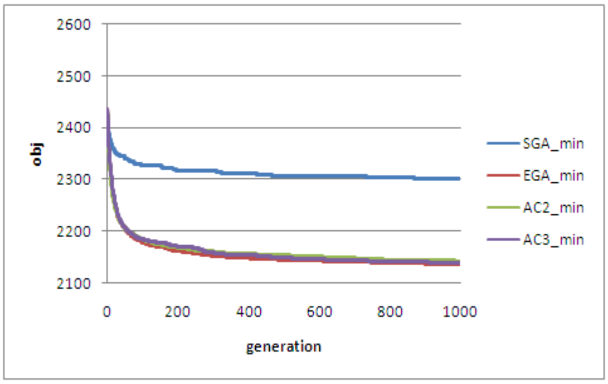

Fig 3. The convergent progress of minimum solution during generation

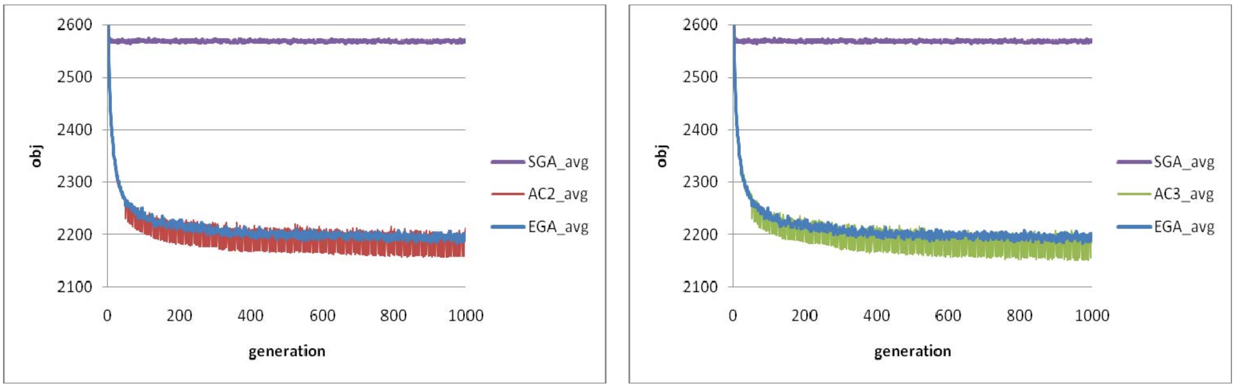

Fig 4. The evolving progress of average solution during generation

From the experimental above, we could find the minimum solution might not be improved too much of our AC2 and AC3 in the small size scheduling problem. But, the average solution will getting variant if AC is injected in the genetic process. It means the chromosomes are not converged into to a local solution while evolving, and we might find a better solution in other complex problem especially in the multi-objective problem.

5. Conclusions

Through this study, we can verify that by combining AC with SPGA, multi-objective scheduling problems can be solved more effectively, especially in the small size problem. In the future, AC with SPGA can be further extended to multidimensional continuous problems, and the procedures of SPMA might be improved to deal with large size problem. Further investigation will be carried out to examine whether it is possible to generate elite chromosomes through better mining algorithms. It is also suggested that different objectives of Flowshop scheduling problems can be further tested such as the arrival time of job is considered, and those with more complex requirements such as sequence dependent setup times.

6. Acknowledgements

This research is supported by grant (NSC 98-2221-E-231 -011) from the National Science Council in Taiwan

7. References

[1] Baker, K.R., Introduction to Sequencing and Scheduling, *Wiley*, New York, 1974.

[2] C. A. Coello, G. T. Pulido, and M. S. Lechuga, Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3), 256–279.

[3] C. R. Reeves, A genetic algorithm for flowshop sequencing. Computer and Operations Research, 1995, 22, 5 – 13.

[4] G. R. Harik, F. G. Lobo, and D. E. Goldberg, The compact genetic algorithm, *IEEE Transactions of Evolution Computing*, 1999, 3 (4), 287 – 297.

[5] P. C. Chang, S. H. Chen, and C. H. Liu, Sub-population genetic algorithm with mining gene structures for multi-objective Flowshop scheduling problems, *Expert Systems with Applications*, 2007, 33, 762–771

[6] P. C. Chang, S. H. Chen, and K. L. Lin, Two phase subpopulation genetic algorithm for parallel machine scheduling problem, *Expert Systems with Applications*, 2005, 29(3), 705–712.

[7] P.C. Chang, Y. W. Wang, and C. H. Liu, New Operators for Faster Convergence and Better Solution Quality in Modified Genetic Algorithm, *Lecture Notes in Computer Science*, 2005, 3611, 983 – 991.

[8] Q. Zhang, J. Sun, and E. Tsang, An evolutionary algorithm with guided mutation for the maximum clique problem, *Evolutionary of Computing*, 2005, 9 (2), 192 – 200.

[9] R. Rastegar, and A. Hariri, A step forward in studying the compact genetic algorithm, *IEEE Transactions of Evolution Computing*, 2006, 14 (3), 277 – 289.

[10] T. P. Bagchi, Multiobjective Scheduling by Genetic Algorithms, *Kluwer academic publishers*, 1999.

[11] Y. W. Wang, C. H. Liu, and C.Y. Fan, Develop a Sub-population Memetic Algorithm for Multi-objective Scheduling Problems, *The 2nd International Conference on Computer and Automation Engineering*, 2010.