A Global Archive Sub-Population Genetic Algorithm with Adaptive Strategy in Multi-objective Parallel-Machine Scheduling Problem

Pei-Chann Chang[1], Shih-Hsin Chen[1], and Jih-Chang Hsieh[2]

[1] Department of Industrial Engineering and Management, Yuan-Ze University, Ne-Li, Tao-Yuan, Taiwan, China
iepchang@saturn.yzu.edu.tw
[2] Department of Finance, Vanung University, Chung-Li 32061, Tao-Yuan, Taiwan, China

Abstract. This research extends the sub-population genetic algorithm and combines it with a global archive and an adaptive strategy to solve the multi-objective parallel scheduling problems. In this approach, the global archive is applied within each subpopulation and once a better Pareto solution is identified, other subpopulations are able to employ this Pareto solution to further guide the searching direction. In addition, the crossover and mutation rates are continuously adapted according to the performance of the current generation. As a result, the convergence and diversity of the evolutionary processes can be maintained in a very efficient manner. Intensive experimental results indicate that the sub-population genetic algorithm combing the global archive and the adaptive strategy outperforms NSGA II and SPEA II approaches.

1 Introduction

Parallel-machine production systems are commonly used in practical manufacturing activities. Parallel machines are able to make the workstations free from being bottlenecks. Regardless of the popularity, the scheduling for parallel machines is still complicated. Garey and Johnson (1979) have shown that two identical parallel machines scheduling with minimizing makespan is NP-hard. Brucker (1998) further indicated that parallel machine scheduling is even strong NP-hard as long as the number of machines is greater than two. The previous works reflect that scheduling for parallel machines is still a great challenge. Because of the NP-hard property, optimality becomes neither effective nor efficient. Many heuristic algorithms were ever proposed for parallel machine scheduling problem such as Hsieh et al. (2003). Among these heuristic algorithms, the performance of genetic algorithms (GA) is convincing and approved by many successful applications such as Neppali et al. (1996), and Sridhar and Rajendran (1996).

Total tardiness time and makespan are considered in this parallel-machine scheduling problem. Total tardiness time reflects if the production meets due-dates and makespan indicates the utilization of the shop floor. Several genetic algorithms

L. Jiao et al. (Eds.): ICNC 2006, Part I, LNCS 4221, pp. 730 – 739, 2006.

have ever been derived for bi-objective or multi-objective optimization problems. Schaffer (1985) proposed vector evaluated genetic algorithm (VEGA), which was the first idea to extend the simple genetic algorithm for multi-objective optimization. Murata and Ishibuchi (1996) proposed multi-objective genetic algorithm (MOGA). MOGA assigns each objective a weight and the weight changes along with the evolving process. Through the weight-changing, MOGA can search Pareto optimal solutions toward different directions. Murata et al. (1996) addressed that MOGA outperforms VEGA on multi-objective flowshop scheduling problem. Zitzler et al. (2002) modified SPEA as SPEA II for multiobjective optimization. Deb et al. (2000) proposed non-dominated sorting genetic algorithm II (NSGA II) by accommodating elitism strategy and crowding distance. Hsieh (2005) proposed grid-partitioned objective space approach based on genetic algorithm with considering multiple objectives. More and more sophisticated genetic algorithms are expected to be developed for solving optimization problems effectively and efficiently.

There are some researchers who propose their subpopulation-like approaches Cochran et al. (2003) proposed a multi-population genetic algorithm (MPGA). Chang et al. (2005) have proposed a two-phase sub population genetic algorithm (TPSPGA) for parallel machine scheduling problems. TPSPGA outperforms NSGA II and MOGA in the numerical experiments. There are still other approaches, such as Segregative Genetic Algorithms (Affenzeller, 2001), Multisexual Genetic Algorithm (Lis and Eiben, 1997), and MO Particle Swam Optimization (Coello et. al, 2004; Mostaghim and Teich, 2004).

Chang et al. (2006) proposed an adaptive multi-objective genetic algorithm for drilling operations scheduling problem in printed circuit board industry. The result indicated that adaptive strategy could be able to improve the solution quality.

Inspired by these pioneer works as discussed above, the SPGA proposed by Chang et al. (2005) is modified by using a global archive Pareto solution and embedded adaptive strategies in this research. In SPGA, the subpopulation works independently; however, according to previous research of these subpopulation algorithms, which create a chance for these subpopulations to be able to exchange information, it may improve the solution quality. Therefore, the modified SPGA considers how to make these subpopulations able to interchange information.

The rest of the research is organized as follows: Section 2 introduces the modified SPGA algorithm, including the global archive technique and adaptive strategies. Because the better parameter settings are not available, the research Design of Experiment is able to obtain better configuration. Then the experimental results of global archive and adaptive strategies are given in Section 3. In addition, the solution of the modified SPGA is compared with the SPGA, NSGA II, and SPEA II. Finally, the conclusion is discussed and the performance of the algorithm is evaluated.

2 Methodology

In this research, using a global archive first modifies SPGA and then an adaptive strategy is embedded in the modified SPGA. The description of SPGA and global archive technique can be found in section 2.1 and the detail procedure is shown in section 2.2. Finally, the adaptive strategies and performance metric are illustrated in section 2.3 and 2.4 respectively.

2.1 The Concept of SPGA and Global Pareto Archive

In order to prevent the searching procedures from being trapped into local optimality, the research applies and modifies the SPGA proposed by Chang et al. (2005). There are two main characteristics of the subpopulation-like method: (1) numerous small sub-populations are designed to explore the solution space; and (2) the multiple objectives are scalarized into a single objective for each sub-population.

In SPGA, each subpopulation works independently and cannot communicate with each other. However, from previous research works of Affenzeller (2001), and Lis and Eiben (1997), the subpopulation should communicate with each other so that it may bring better convergence and diversity. Therefore, this research uses a global archive to exchange information of better solutions among sub-populations while they are exploring the different solution space together. The framework of the global archive is shown as in Fig 1.

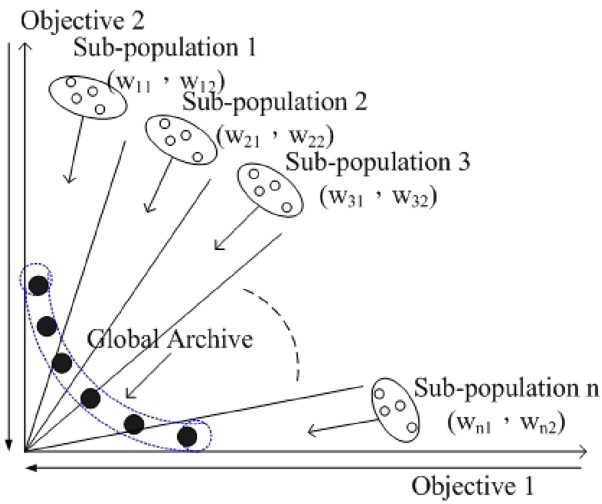

Fig. 1. The framework of the global archive for SPGA

2.2 Procedures of Modified Sub-Population Genetic Algorithm

Because SPGA does not share the Pareto archives with each other, it might lose the chance to obtain a better solution; the other sub-populations cannot apply it to improve the solution quality. Consequently, the main idea of the modified SPGA is to collect Pareto optimal solutions from sub-populations as a global Pareto archive. The global Pareto archive is expected to improve the solution quality and maintain the diversity.

The algorithmic procedure of the modified SPGA is explained in the following:

Algorithm: The modified SPGA()

Initialize()
DividePopulation()
AssignWeightToEachObjectives()
counter ← 0
while counter < Iteration **do**
    for i = 1 to *ns* **do**
        Selection and *Elitism*(i)
        Crossover(i)
        Mutation (i)
        EvaluateSolutions(i)
        Fitness(i)
        UpdateglobalParetoarchive()
        Replacement(i)
    end for
    counter ← counter + 1
end while
exit

The procedure *initialize* is used to generate the chromosomes of a population, whose size is determined by user. The procedure *DividePopulation* is to divide the original population into *ns* sub-populations.

At the procedure *AssignWeightToEachObjectives*, each sub-population is assigned different weight values and the individuals in the same sub-population share the same weight value. Because the research focuses on bi-criteria problem, the vector size is two. The equation of combination of weight value below:

$$(W_{n1}, W_{n2}) = \left(\frac{1}{N_s+1}\cdot n\,,\ 1-\frac{1}{N_s+1}\cdot n\right) \tag{1}$$

where n is the nth sub-population.

After the weight value assignment, the corresponding scalarized objective value of the two objectives in sub-population can be written as equation 2.

$$f(x) = W_{n1}\cdot Z_{TT}(x) + W_{n2}\cdot Z_{TC}(x) \tag{2}$$

where Z_{TT} and Z_{TC} denote total tardiness time and makespan for each solution x.

Because the scales of the two objectives are different, the objective values are normalized in a unit interval. The *Elitism* strategy at the first stage randomly selects a number of individuals from non-dominated set into mating pool, so that individuals can be selected while the *crossover* procedure. The *Elitism* strategy for global archive SPGA is to collect best non-dominated solution from all subpopulations and it copies a proportional elites into the selection procedure. The binary tournament selection is employed in the *selection* operation. The smaller objective value has better chance to be selected. Besides, it also employs some elites from the global archive. Finally, the replacement strategy is the total replacement one, which means the offspring substitutes the parent solution entirely.

2.3 Procedure of the Adaptive Sub-Population Genetic Algorithm

The procedure of the adaptive SPGA includes measure diversity of the population and the adaptive crossover and adaptive mutation operator will apply the result into their own operations.

Two adaptive strategies are embedded into the modified SPGA. The first one and the second one were proposed by Srinivas and Patnaik (1994), and the Zhu and Liu (2004) respectively.

The method of Srinivas and Patnaik (1994) has been widely applied, which depends on the fitness judgment and the fitness normalization. The goodness of a solution is judged by the average fitness value. If the smaller fitness value means a better solution, the definition of better solution here is the solution whose fitness value is lower than the average fitness. Thus, these solutions apply smaller crossover rate and mutation rate. The scale of the probability is based on the normalization ratio among solutions. On the other hand, the worsen solution is mated or mutated in a higher probability. The adaptive crossover and mutation operators are as equation (3) and (4) respectively:

$$p_c = \begin{cases} k_1 (f_i - f_{\min}) / (f_{\max} - \overline{f}) & if \quad f_i \leq \overline{f} \\ k_3 & otherwise \end{cases} \tag{3}$$

$$p_m = \begin{cases} k_2 (f_i - f_{\min}) / (f_{\max} - \overline{f}) & if \quad f_i \leq \overline{f} \\ k_4 & otherwise \end{cases} \tag{4}$$

Another adaptive strategy proposed by Zhu and Liu (2003) consists of three steps the distance measure, diversity measure, and diversity control, which can be expressed by the following equation (5), (6), and (7):

$$\text{Step 1: } Ham(\mathrm{x}, \mathrm{y}) = \sum_i | \operatorname{sgn}(x[i] - y[i]) | \tag{5}$$

$$\text{Step 2: } gD(P) = \frac{1}{2} \sum_{i \neq j} Ham(P[i], p[j]) \tag{6}$$

$$\text{Step 3: } p' = \max(p_{min}, \min(p_{max}, p \left[1 + \frac{\xi(gD_t - gD)}{gD} \right])) \tag{7}$$

where gD_t : The target population diversity.

In equation (5), the distance measure applies the hamming distance between two solutions. The diversity measure evaluates the diversity of all solutions in equation (6). It detects how the “health level” of the population. Finally, the diversity control is to modify the rate according to the target population diversity. If the population diversity is higher than the target diversity, the rate is decreased. Otherwise, the rate is increased.

2.4 Evaluation Metric

The research uses $D1_R$ to evaluate the solution quality. Knowles and Corne (2002) indicated that $D1_R$ considers the convergence and diversity at the same time. After a run, an algorithm obtains a set of Pareto solutions, which is compared with a reference set. Thus, the $D1_R$ value is obtained. The lower $D1_R$ value, the better the solution quality. Therefore, the $D1_R$ provides a basis for comparing the performance among different algorithms in the study. The equation of $D1_R$ is represented as equation (8) and (9).

$$D1_R(A_j) = \frac{1}{\left|Z^*\right|} \sum_{y \in Z^*} \min\{d_{xy} \mid x \in A_j\} \tag{8}$$

$$dxy = \sqrt{\sum_{i=1}^{n} (f_i^*(y) - f_i(y))^2} \tag{9}$$

where A_j: A set of Pareto solution obtained by an algorithm

Z^*: The reference solution or true Pareto solution

$\left|Z^*\right|$: The number of reference solution

3 Numerical Experiments

The data collected from a printed circuit board factory are applied to be the test instances[1]. Three job/machines combinations are considered, i.e., 35/10, 50/15, 65/18.

3.1 Experiment Design for Parameters Settings in SPGA

The subsection tries to determine the optimal parameter setting in the algorithms. Then, the next experiment applies the result of these parameter settings and compares it with NSGA II and SPEA II.

There are several parameters that may influence the performance of the algorithm. For example, the larger population size may find better solution quality but cost higher computational expense. When the number of sub-populations is larger, it may have better diversity. However, it may also be a trade-off that to reduce the number of generations. Moreover, the secondary crossover and mutation operator are also considered because it may provide better solution quality. The crossover rate and mutation rate are set to 0.9 and 0.1 respectively. The factors and treatments of these factors are as shown in Table 1. The detail ANOVA result can be obtained at our website and the suggested parameter settings is presented in Table 2.

[1] The data can be assessed at our website: http://ppc.iem.yzu.edu.tw/download.html

Table 1. The default parameter setting and the treatments of different factors

Factor	Treatment
Number of job (A)	35/10, 50/15, 65/18 (jobs/ machines)
Number of sub-population (B)	10, 20, 30, 40
Population Size (C)	100, 155, 210
Secondary Crossover Operator (D)	Apply multiple crossover (1), not using it (0)
Secondary Mutation Operator (E)	Apply multiple mutation (1), not using it (0)

Table 2. The suggested parameter for the modified SPGA

Factor	Treatment
Crossover Rate	0.9
Mutation Rate	0.1
Population Size	210
Number of sub-population	40

3.2 Comparisons for Adaptive Strategies

The experiment compares different adaptive strategies, including the method of Srinivas and Patnaik (1994) and the adaptive strategy of Zhu and Liu (2004). They are coded as 0, 1, and 2 in the experiment. The ANOVA table is available on our website and it represents the interaction between the instance and method which causes significant difference. Therefore, the interaction plot of the two factors is depicted in Fig 2. It shows that the adaptive strategies do not perform better in small size instances, while they outperform in large size instances. Duncan grouping method is

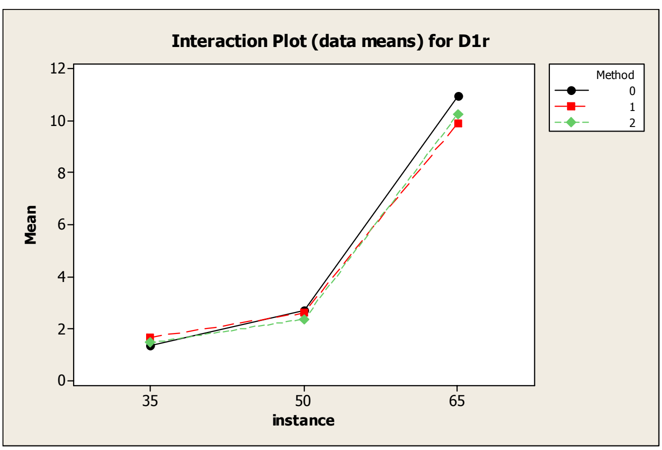

Fig. 2. The interaction plot between the instance and methods

Duncan Grouping	Mean	N	Method
A	6.8502	60	0
B	6.3144	60	2
B	6.2674	60	1

Fig. 3. The Duncan grouping method

applied to distinguish the group for the adaptive strategies. The grouping result is shown in figure 3. It shows that the modified SPGA with the adaptive strategies are better than the modified SPGA without adaptation. Since the time-complexity of Zhu and Liu (2004) is higher than Srinivas and Patnaik (1994), the study suggests using the later one when researchers would like to apply the adaptive strategy.

3.3 Numerical Results

After the study obtains the result of adaptive strategy for SPGA, the section compares the result with Modified SPGA, NSGA II and SPEA II, by three testing instances. Table 3 shows the statistics result of instances of 35 jobs and 10 machines, 50 jobs and 15 machines, and 65 jobs and 18 machines.

From the three instances, the modified SPGA is superior to the SPGA, NSGA II, and SPEA II in minimum, average, and maximum value. There is only one exception that the maximum value of modified SPGA is not better than SPGA and SPEA II in the instance of 65 jobs and 18. Then, the adaptive SPGA is better than the Modified SPGA 5.07% through the three instances.

Table 3. The min, average, and max value of different algorithms of the three instances

Instance	Algorithm	Min	Avg.	Max
35/10	Adaptive SPGA	0.494	1.667	3.391
	Modified SPGA’	0.56	1.4722	2.5147
	NSGA II	5.16	11.82	22.22
	SPEA II	4.8	10.39	22.48
50/15	Adaptive SPGA	1.418	2.609	3.554
	Modified SPGA’	1.72	2.876	3.901
	NSGA II	9.68	11.74	13.79
	SPEA II	7.65	10.27	12.89
65/18	Adaptive SPGA	5.092	9.925	13.192
	Modified SPGA’	7.537	10.611	13.941
	NSGA II	20.97	23.08	25.43
	SPEA II	7.7	10.3	12.9

4 Conclusion and Future Works

A modified SPGA and an adaptive SPGA were proposed for solving parallel machine scheduling problem with minimizing total tardiness time and makespan. Production data collected from a printed circuit board factory were applied as test instances. The numerical result indicated that SPGA with adaptive strategy perform better in large size test instances than SPGA without adaptation. Two genetic algorithms for multi-objective optimization, NSGA II and SPEA II, were compared with the proposed methods in this research. Extensive studies were conducted and the result reported that the adaptive SPGA and modified SPGA proposed in this research tend to outperform NSGA II and SPEA II especially in the large size problems. This also means the adaptive SPGA and modified SPGA are potential in the future works.

In the future research, although the algorithm is attractive to implement MO problem, we can still consider to combine SPGA with local search algorithms that may bring better solution quality.

References

1. Affenzeller, M. (2001). New Generic Hybrids Based Upon Genetic Algorithms. *Institute of Systems, Science Systems Theory and Information Technology*, Johannes Kepler University.
2. Brucker, P. (1998). *Scheduling Algorithm*. Berlin: Springer.
3. Chang, P.C., Chen, S.H., Lin, K.L. (2005). Two-phase sub population genetic algorithm for parallel machine-scheduling problem. *Expert Systems with Applications*, 29 (3), 705-712.
4. Chang, P.C. Hsieh, J.C., Wang, C.Y. (2006). Adaptive multi-objective genetic algorithms for scheduling of drilling operation in printed circuit board industry. To appear in *Applied Soft Computing*.
5. Cochran, J.K., Horng, S., Fowler, J.W. (2003). A multi-objective genetic algorithm to solve multi-objective scheduling problems for parallel machines. *Computers and Operations Research*, 30, 1087-1102.
6. Coello, C.A.C., G.T. Pulido, M.S. Lechuga (2004), “Handling multiple objectives with particle swarm optimization,” IEEE Transactions on Evolutionary Computation, 8(3), 256- 279.
7. Deb, K., Amrit Pratap, S.A., Meyarivan, T. (2000). A fast and elitist multi-objective genetic algorithm-NSGA II. *Proceedings of the Parallel Problem Solving from Nature VI Conference*, 849-858.
8. Garey, M.R., Johnson, D.S. (1979). *Computers and Intractability*: *A guide to the theory of NP-completeness*. San Francisco, CA: New York Freeman.
9. Hsieh, J.C. (2005). Development of grid-partitioned objective space algorithm for flowshop scheduling with multiple objectives. Proceedings of the the 6th Asia Pacific Industrial Engineering and Management Systems Conference 2005.
10. Hsieh, J.C., Chang, P.C., Hsu, L.C. (2003). Scheduling of drilling operations in printed circuit board factory. *Computers and Industrial Engineering*, 44(3), 461-473.
11. Knowles, J.D. and Corne, D.W. (2002). On metrics for comparing non dominated sets. *Proceedings of the 2002 congress on evolutionary computation conference* (*CEC02*), 711-716. New York: IEEE Press.

12. Lis, J. and Eiben, A.E. (1997). A multisexual Genetic Algorithm for multicriteria optimization. *In Proceedings of the 4th IEEE Conference on Evolutionary Computation*, 59-64.
13. Murata, T., Ishibuchi, H. (1996). MOGA: Multi-objective genetic algorithm. *Proceedings of the Second IEEE International Conference on Evolutionary Computation*, 170-175.
14. Murata, T., Ishibuchi, H., Tanaka, H. (1996). Genetic algorithm for flowshop scheduling problem. *Computers and Industrial Engineering*, 30, 1061-1071.
15. Mostaghim , S., Teich, J. (2004), "Covering Pareto-optimal fronts by subswarms in multi-objective particle swarm optimization," *Evolutionary Computation*, 2, 1404 - 1411.
16. Neppali, V.R., Chen, C.L., Gupta, J.N.D. (1996) Genetic algorithms for the two-stage bicriteria flowshop problem. *European Journal of Operational Research*, 95, 356-373.
17. Schaffer, J.D. (1985) Multiple objective optimization with vector evaluated genetic algorithms. *Proceedings of First International Conference on Genetic Algorithms*, 93-100.
18. Sridhar, J., Rajendran, C. (1996). Scheduling in flowshop and cellular manufacturing systems with multiple objectives – a genetic algorithm approach. *Production Planning and Control*, 7, 374-382.
19. Srinivas, M. and Patnaik, L. M. (1994). Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms, *IEEE Transactions on Systems, Man and Cybernetics*, 24 (4), pp. 656-667.
20. Zhu, K.Q., Liu, Z. (2004). Population diversity in permutation-based genetic algorithm. *Proceedings of the 15th European Conference on Machine Learning*, ECML 2004, Pisa, Italy. Springer. 537--547.
21. Zitzler, E., Laumanns, M., Thiele, L. (2002). SPEA 2: improving the strength Pareto evolutionary algorithm for multiobjective optimization, *Evolutionary Methods for Design, Optimisation and Control*, 1-6, Giannakoglou, K., Tsahalis, D., Periaux, J., Papailiou, K., and Fogarty, T. eds. CIMNE, Barcelona, Spain.