Multi-Niche Crowding in the Development of Parallel Genetic Simulated Annealing

Zhi-Gang Wang M. Rahman Yoke-San Wong
Depart of mechanical engineering, National University of Singapore, 10 Kent Ridge Crescent, Singapore, 119260
+65-68744644 +65-68742168 +65-68742221
wangzhigang@nus.edu.sg mpemusta@nus.edu.sg mpewys@nus.edu.sg

ABSTRACT
In this paper, a new hybrid of genetic algorithm (GA) and simulated annealing (SA), referred to as GSA, is presented. In this algorithm, SA is incorporated into GA to escape from the local optima. Then, the idea of hierarchical parallel GA is borrowed to parallelize GSA for the optimization of multimodal functions. In addition, multi-niche crowding is used to maintain the diversity in the population of parallel GSA. The performance of the proposed algorithms is evaluated against a standard set of multimodal benchmark functions. Multi-niche crowding PGSA and normal PGSA show some remarkable improvement in comparison with the conventional parallel GA and the breeder genetic algorithm.

Categories and Subject Descriptors
[**Genetic algorithm**]

General Terms
Algorithms

Keywords
Genetic algorithm, simulated annealing, parallel genetic algorithm.

1. INTRODUCTION
When applying GA to multimodal optimization problems, the successful application depends on the preservation of good individuals into next generation and the maintaining of diversity of individual solutions in the search space. SA is an effective way to preserve good individuals into the next generation, and crowding strategy is an alternative approach to maintain the population diversity and postpone premature convergence.

This paper presents a new hybrid of GA and SA, referred to as genetic simulated annealing (GSA). GSA is also parallelized to find multiple optimal solutions for the multimodal functions. More importantly, multi-niche crowding strategy is used to maintain population diversity of parallel GSA (PGSA). The paper aims to demonstrate that multi-niche crowding PGSA is a powerful optimization strategy.

2. THE MNC-PGSA ALGORITHM
2.1 Genetic Simulated Annealing
After crossover and mutation for a couple of individuals, there are four chromosomes. In conventional GA, two parents are replaced by their offspring. But in GSA, two chromosomes are chosen to form the next generation from these four individuals. The selection criterion is based on the fitness values of these four individuals. Individuals with higher fitness values have a greater probability of surviving into the next generation. Those with less fitness values are not necessarily discarded. Instead, local selection strategy of SA is applied to select them with a probability related to the current temperature (as in SA).

Initially, the mutation probability p_m of GSA is set to a higher value, and a simple annealing process is then used to adjust p_m. After every ten generations, p_m is updated with $p_m \times \alpha$ until it reaches to a certain value, where α is the cooling rate of SA. Thus, at the initial stage, when manipulating the cooling schedule of SA properly, the initial higher temperature can ensure that parents will be replaced by their offspring after mutation and crossover whether they are much fitter or not. More importantly, the initial higher mutation probability is capable of improving population diversity greatly, which can eliminate the premature convergence problem of conventional GA. On the contrary, at the later stage the mutation probability and the temperature become lower, and the chances for the fitter parents to be replaced decrease greatly. In this way, the current best individuals may always remain in the next generation. Thus, the possibility of removing potentially useful individuals in the last generation because of the mutation operation can be reduced.

In addition, good parallelizable property of GA is applied to parallelize GSA. A master-slave/coarse-grained PGA, which combines master-slave PGA and coarse-grained PGA together, has been used to parallelize GSA. Thus, GSA shows tighter coupling of GA and SA as SA controls a number of distinct GAs running in parallel.

In this study, optimization problems of real-valued functions are considered. And the real-value coding scheme is employed to represent the chromosome. In this study, at the initial stage, uniform distributed mutation is used. When the decreasing rate of the average fitness values is less than 0.01, Gaussian mutation is used. In the reproduction process of PGSA, tournament selection approach is used to select individuals for the next generation. In tournament approach, a sub-group is initially selected randomly from the population. Then, a 'tournament' competition is taken place, and the winner is inserted into the next population.

2.2 Multi-niche Crowding in PGSA
There are two steps for this crowding selection approach. Firstly, an individual A is selected for mating. Secondly, its mate M is

GECCO'05, June 25-29, 2005, Washington, DC, USA.
ACM 1-59593-010-8/05/0006.

chosen with the crowding selection instead of the fitness proportionate reproduction (FPR) of SGA. Based on the similarity to A, M is selected from a randomly chosen group of C_s individuals. After picking the mate of A, the genetic operators of crossover and mutation are then applied; one pair of offspring is generated. For each of these two offspring, MNC is again used to select an individual from the population for replacement by this offspring. During the replacement step, a replacement policy called worst among the most similar is used in MNC [1].

MNC has been considered to maintain population diversity for PGSA. In the implementation, MNC was only applied among half the number of slave processors to select mating individuals and the individuals to be replaced by offspring. On the other half number of slave processors, normal PGSA is used in order to maintain the good local selection ability of SA. With the incorporation of MNC into PGSA, a new algorithm, called MNC-PGSA, is obtained. This MNC-PGSA algorithm maintains good population diversity and inherits good convergence from SA.

3. RESULTS AND DISCUSSION

Three multimodal benchmark functions have been used to compare the performance of MNC-PGSA and PGSA with other algorithms, as shown in Figures 1, 2 and 3. For Rastrgin and Schwefel's functions, it can be seen that the number of function evaluations using MNC-PGSA and PGSA is much smaller than that using PGA. PGSA can maintain a good diversity with a higher mutation probability at the initial stage; it can eliminate premature convergence to suboptimal minima. At the later stage, the local selection strategy of SA can ensure that best solutions are not discarded after crossover and mutation operator. Therefore, PGSA can approach or converge on the global minimum. Except the advantages of PGSA, MNC-PGSA can maintain good population diversity. The performance of MNC-PGSA and PGSA gets better with the higher problem size n.

Griewank's function is regarded as one of the most difficult test functions. An average of 59520 evaluations is needed to solve this problem by Mühlenbein *et al* [2]. MNC-PGSA and PGSA found the minimum values (< 0.001) of Griewank's function with less than half of the number of function evaluations using PGA, as shown in Figure 3. More importantly, in 50 runs, these minimum values were found. Therefore, MNC-PGSA and PGSA is able to obtain much better solutions with a higher convergence speed than PGA. Figure 3 shows that less number of function evaluations with MNC-PGSA was needed to converge to global optima for Function F3 with the problem size of 10 than that those for PGSA. Thus, MNC-PGSA performs better than PGSA.

In order to compare the efficiency of PGSA with that of BGA [3], the same termination criterion as that for BGA was used for Griewank's function. The computation results are listed in Figure 3. Figure 3 shows that the performance of MNC-PGSA is better than that of BGA. And MNC-PGSA performs better with the larger problem size. Thus, MNC-PGSA has a good scalability. The difference in performance between MNC-PGSA and PGSA for solving Griewank's function is much more obvious than that for solving Rastrgin and Schwefel's functions with these two algorithms. Because it is much more difficult to obtain the global optima of Griewank's function, MNC-PGSA shows its high efficiency at this situation. This reiterates that MNC-PGSA is a powerful optimization method in comparison to PGSA and BGA.

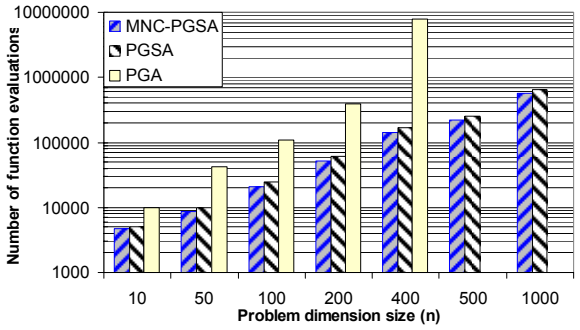

Figure 1. Performance comparison between PGA [2], PGSA and MNC-PGSA for Rastrigin's Function

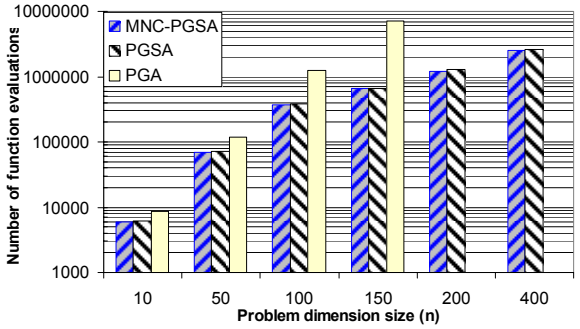

Figure 2. Performance comparison between PGA [2], PGSA and MNC-PGSA for Schwefel's Function

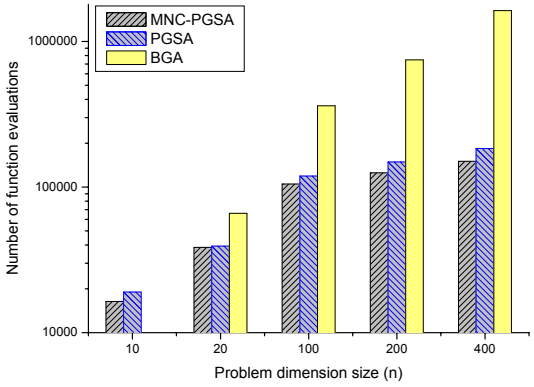

Figure 3. Performance comparison between BGA [3], PGSA and MNC-PGSA for Griewank's Function

4. REFERENCES

[1] Cedeno, W. and Vemuri, V.R. Analysis of speciation and niching in the multi-niche crowding GA. *Theoretical Computer Science*, 229, 1-2 (1999), 177-197.

[2] Mühlenbein, H., Schomisch, M. and Born, J. The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17(1991), 619-632.

[3] Mühlenbein, H. and Schlierkamp-Voosen, D. Predictive models for the breeder genetic algorithm I. Continuous parameter optimization. *Evolutionary Computation*, 1(1993), 25-49.