Distributed NSGA-II Sharing Extreme Non-dominated Solutions

Yuji Sato
Hosei University
3-7-2, Kajinocho, Koganei-shi
Tokyo, Japan 184-8584
yuji@k.hosei.ac.jp

Mikiko Sato
Tokai University
2-3-23, Takanawa, Minato-ku
Tokyo, Japan 108-8619
mikiko.sato@tokai.ac.jp

Minami Miyakawa
Hosei University
3-7-2, Kajinocho, Koganei-shi
Tokyo, Japan 184-8584
miyakawa@cis.k.hosei.ac.jp

ABSTRACT

A recent trend in multiobjective evolutionary algorithms is to increase the population size to approximate the Pareto front with high accuracy. On the other hand, the NSGA-II algorithm widely used in multiobjective optimization performs non-dominated sorting in solution ranking, which means an increase in computational complexity proportional to the square of the population. This execution time becomes a problem in engineering applications. It is also difficult to achieve high speeds while maintaining the accuracy of solution searching by simply applying fast, parallel processing to standard genetic operations. In this paper, we propose NSGA-II distributed processing in a many-core environment and a migration method that shares extreme Pareto solutions of the current generation among all cores after performing compensation of the non-dominated solution set obtained by distributed processing. Using a two-objective and three-objective constrained knapsack problem for evaluation, we show that the proposed method is effective in improving diversity in solution searching while shortening execution time and increasing the accuracy of solution searching.

CCS CONCEPTS

•**General and reference** → *Experimentation; Design; Performance; Verification;* •**Computing methodologies** → *Parallel computing methodologies;*

KEYWORDS

multiobjective evolutionary algorithms, NSGA-II, distributed processing, many-core environment

1 DISTRIBUTED NSGA-II SHARING EXTREME NON-DOMINATED SOLUTION

In recent years, the trend in multiobjective evolutionary algorithms has been to increase the population size to approximate the Pareto-optimal front [2] with high accuracy. Increasing the population size, however, results in an exponential increase in the computational complexity required for evaluating the dominant-subordinate relationships among solutions. As a result, execution time can be a problem when applying such an approach to engineering applications. However, high-accuracy multiobjective evolutionary algorithms such as NSGA-II [2], SPEA2 [5], MOEA/D [4], and NSGA-III [1] add original processing different from ordinary genetic operations to improve convergence and the diversity of the nondominated solution set, which reflects the fact that simply applying the technologies of prior research cannot maintain the accuracy of solution searching. For example, when executing nondominated sorting — a feature of NSGA-II — on multiple islands in a divided manner and evaluating the elite individuals (nondominated solutions) on each island across the entire population, the problem arises that some of those solutions may not be nondominated after all.

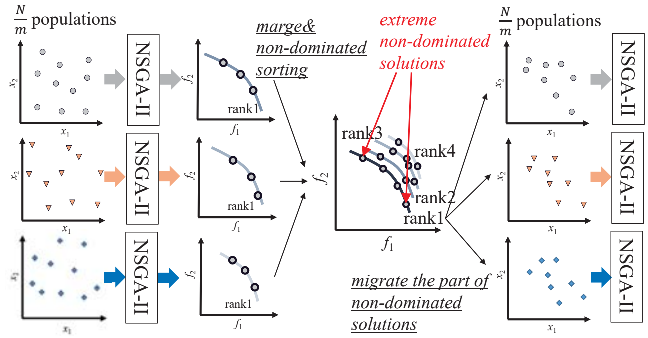

Figure 1: Concept of DNSGA-II with migration.

To resolve the issue, we have already proposed the method "DNSGA-II [3]" for achieving fast, parallel processing of NSGA-II while maintaining the accuracy of the Pareto-optimal front. Here, we propose the newly migration method sharing extreme non-dominated solutions is shown in Fig. 1. In the following, we denote DNSGA-II using this migration method as Distributed NSGA-II sharing extreme non-dominated solutions (e-DNSGA-II). In the figure, the proposed migration method sharing extreme non-dominated solutions first gathers the non-dominated solution sets (rank 1 solution sets) obtained by solution searching on each CPU and again performs non-dominated sorting and compensation processing with ranking. It then preferentially allocates the non-dominated solutions at both ends of the current Pareto front to all CPUs in place of the excluded false non-dominated solutions. In the event that all non-dominated solutions are true solutions after again applying non-dominated sorting, the method deletes those individuals with small crowding distance (CD) values replacing them with these non-dominated solutions at both ends of the Pareto front. In short, this method performs migration by deleting false non-dominated solutions and non-dominated solutions with small CD values and replacing them with extreme non-dominated solutions at both ends of the current Pareto front (shares extreme non-dominated solutions among all islands).

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan
 978-1-4503-5764-7/18/07…$15.00
DOI: 10.1145/3205651.3208769

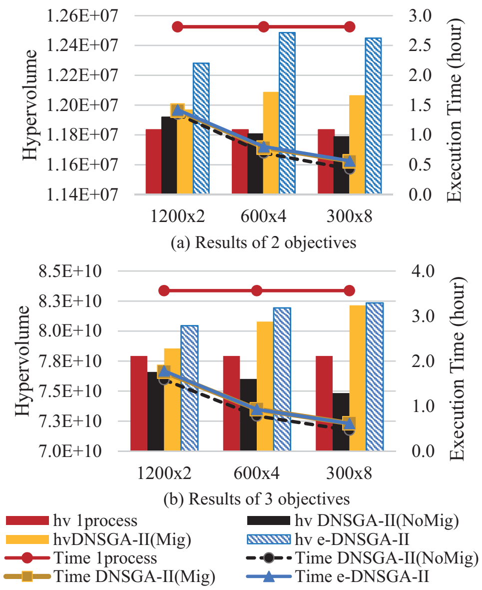

Figure 2: Hypervolume and execution time of each method.

2 EVALUATION AND DISCUSSION

Using the constrained knapsack problem, we performed an evaluation with respect to discrete optimization problems. The problem has n items and k knapsacks. Considering that using this problem for an evaluation takes time and that a general island model has many design variables, we here perform a comparison evaluation targeting two items — hypervolume and execution time — for the case of executing conventional NSGA-II on a single CPU and the case of parallel execution of NSGA-II using the proposed method.

(A) Experiments comparing hypervolume and execution time

For the sake of brevity, we here examine the case for a migration interval of 300. The same behavior was observed for the other migration intervals. Execution results for NSGA-II executed on a single CPU (1process), DNSGA-II performing no migration (DNSGA-II(NoMig)), DNSGA-II (DNSGA-II(Mig)) and e-DNSGA-II (e-DNSGA-II) are shown in Fig. 2 for two objectives and three objectives, respectively. The left and right axes in the figures show hypervolume and execution time, respectively. For the case of two objectives in Fig. 2, results reveal no major differences among the various methods for 2 CPUs, but for 4 and 8 CPUs, DNSGA-II and the e-DNSGA-II method proposed here achieve a high hypervolume. Moreover, for 4 CPUs, e-DNSGA-II achieves a particularly high hypervolume. Similarly, for the case of three objectives in Fig. 2, the results for DNSGA-II and e-DNSGA-II show high hypervolume. Additionally, for no migration, results for both two and three objectives reveal that search accuracy (hypervolume) drops with increase in parallelism. We consider the reason for this to be that increasing the number of islands to accelerate processing increases the frequency of appearance of solution candidates erroneously classified as nondominated solutions.

Next, for the case of executing NSGA-II on a single CPU, an optimal hypervolume could not be obtained even after eight hours. As a result, the execution time of NSGA-II is approximately 9 times and 13 times that of e-DNSGA-II for achieving a suboptimal solution with a hypervolume of 11000000 and 7.5E+10 for two and three objectives, respectively. These results reflect the high-speed operation of the proposed e-DNSGA-II method.

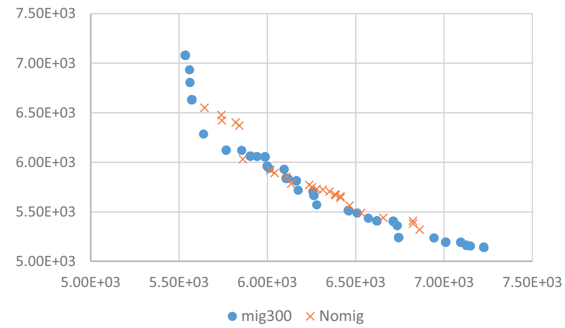

Figure 3: Comparison of NSGA-II and e-NSGA-II Pareto fronts for 2 objectives.

(B) Comparison of diversity and convergence by shape of Pareto-optimal front

The shapes of the Pareto-optimal front for e-DNSGA-II and single-CPU NSGA-II are compared in Fig. 3 for two objectives. The proposed e-DNSGA-II method achieves greatly improved diversity at both extremes of the Pareto-optimal front. It can also be observed from these results that the final number of non-dominated solutions increases from the 24 of NSGA-II to the 39 of e-NSGA-II and that the latter features a uniform distribution of non-dominated solutions.

In the evaluation using the constrained knapsack problem, it was seen for the case of no migration that search accuracy tended to drop by distributed processing. We explain the reason for this in conjunction with the evaluation results using real-valued functions. We found in this evaluation that an improvement in diversity generally came at the expense of a drop in convergence ability and that the hypervolume value that was eventually obtained tended to drop. On the other hand, we consider that applying appropriate migration as in the case of DNSGA-II and e-DNSGA-II has the effect of compensating for this drop in convergence ability and that, in addition to improving diversity, has the capability of simultaneously achieving high-speed parallel processing and improving search ability (improving hypervolume) in the end.

REFERENCES

[1] K. Deb and H. Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (Aug 2014), 577–601.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 2 (Apr 2002), 182–197. https://doi.org/10.1109/4235.996017

[3] Y. Sato, M. Sato, and M. Miyakawa. 2017. Distributed NSGA-II using the divide-and-conquer method and migration for compensation on many-core processors. In *2017 21st Asia Pacific Symposium on Intelligent and Evolutionary Systems (IES)*. 83–88.

[4] Q. Zhang and H. Li. 2007. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation* 11, 6 (Dec 2007), 712–731.

[5] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. 2001. *SPEA2: Improving the Strength Pareto Evolutionary Algorithm.* Technical Report. 95–100 pages.