Distributed NSGA-II with Migration using Compensation on Many-core Processors for Improving Performance and Accuracy

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 nondominated solution set with high accuracy. And the execution time becomes a problem in engineering applications. In this paper, we propose distributed, high-speed NSGA-II using a many-core environment to obtain a Pareto-optimal solution set excelling in convergence and diversity. This method improves performance while maintaining the accuracy of the Pareto-optimal solution set by repeating NSGA-II distributed processing in a many-core environment inspired by the divide-and-conquer method together with migration processing for compensation of the nondominated solution set obtained by distributed processing. On comparing with NSGA-II executing on a single CPU and parallel, high-speed NSGA-II using a standard island model, it was found that the proposed method greatly shortened the execution time for obtaining a Pareto-optimal solution set with equivalent hypervolume while increasing the accuracy of solution searching.

CCS CONCEPTS

•General and reference \rightarrow Experimentation; Design; Performance; Verification; •Computing methodologies \rightarrow Parallel computing methodologies;

KEYWORDS

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

1 DISTRIBUTED NSGA-II WITH COMPENSATION ON MANY-CORE PROCESSORS: DNSGA-II

In recent years, the trend in multiobjective evolutionary algorithms has been to increase the population size to approximate the Paretooptimal front [2] with high accuracy [3]. 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

GECCO '17 Companion, Berlin, Germany



Figure 1: Concept of DNSGA-II with migration.

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. The standard island model repeats the process of dividing the population into subgroups (islands), executing NSGA-II in parallel, and migrating elite individuals on each island to other islands at appropriate times. However, 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. In addition, increasing the number of islands to accelerate processing increases the frequency of appearance of solution candidates erroneously classified as nondominated solutions.

To resolve the issue, we propose a method for achieving fast, parallel processing of NSGA-II while maintaining the accuracy of the Pareto-optimal front. As shown in Fig. 1, the proposed **DNSGA-II** method executes NSGA-II in each subgroup in parallel, gathers the nondominated solution sets (rank 1 solution sets) obtained by solution searching on each CPU, and again performs nondominated sorting with ranking as compensation processing. Next, the method performs migration by allocating to each CPU a portion of the nondominated solution set obtained by compensation. It then proceeds to the next generation of solution searching by NSGA-II on each CPU. This repeated execution of NSGA-II distributed processing in a many-core environment while performing compensation processing of the false nondominated solution sets obtained in each subgroup achieves high-speed NSGA-II while maintaining the accuracy of solution searching.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

^{© 2017} Copyright held by the owner/author(s). 978-1-4503-4939-0/17/07...\$15.00 DOI: http://dx.doi.org/10.1145/3067695.3075974

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany

Yuji Sato, Mikiko Sato, and Minami Miyakawa

2 EVALUATION AND DISCUSSION

To assess the effectiveness of the proposed method, we compared two items — the accuracy of solution searching and execution time among three types of NSGA-II execution: conventional NSGA-II on a single CPU, parallel execution of NSGA-II applying the standard island model, and parallel execution of NSGA-II by the proposed model. We used hypervolume as an indicator of the accuracy of solution searching, and for test problems, we used the multiobjective optimization problems taken from the problems attached to the NSGA-II source code. The test execution environment is Intel Xeon X5680, dual processor system (12 cores). Experimental results were taken to be the average of 10 trials.

As represented in Fig. 2, in all test problems, performing migration with the proposed method tended to result in a higher hypervolume. And, with the exception of performing migration generation, the proposed method was able to keep execution time below that of single-CPU NSGA-II (Time-1process) despite the overhead associated with migration processing. On the other hand, the results for the conventional island model in all test problems showed no significant difference with the hypervolume values obtained by the proposed method for a small degree of parallelism (2-4 CPUs). On increasing the degree of parallelism, however, islandmodel results converged to hypervolume values less than those of the proposed method. Furthermore, for the CTP1 test problems, the proposed method could surpass the accuracy of solution searching by a single CPU by shortening the migration interval with a high degree of parallelism such as 12 or 8 CPUs. It was also found that setting an appropriate migration interval could result in even higher hypervolume values.

We evaluated the approximate accuracy of the Pareto-optimal front of nondominated solution sets in terms of execution time for the test problems. As represented in Fig. 3, in the case of DNSGA-II, we set the population per CPU to 200 individuals and performed the evaluation while increasing the degree of parallelism and total population. Referring to these figures, (a) shows bar graphs and plots for hypervolume and execution time, respectively, versus total population and (b) shows the relationship between hypervolume and execution time for each value of total population. The results in (b) show that the execution time required for obtaining equivalent hypervolume is significantly shorter by parallel DNSGA-II than single-CPU NSGA-II. When increasing the degree of parallelism while keeping the population per CPU fixed, it was found that the same hypervolume values as obtained by NSGA-II executed on a single CPU could be obtained, and in addition, that execution time could be shortened from 10 times to a maximum of 60 times for the same hypervolume value.

ACKNOWLEDGMENTS

This research is partly supported by the collaborative research program 2016, Information Initiative Center, Hokkaido University.

REFERENCES

 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.







(b) Relationship between hypervolume and execution time

Figure 3: Evaluation results of hypervolume and execution time in CTP5.

- [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. DOI: http://dx.doi.org/10.1109/4235.996017
- [3] H. Ishibuchi, N. Akedo, and Y. Nojima. 2015. Behavior of Multiobjective Evolutionary Algorithms on Many-Objective Knapsack Problems. *IEEE Trans*actions on Evolutionary Computation 19, 2 (April 2015), 264–283. DOI:http: //dx.doi.org/10.1109/TEVC.2014.2315442
- [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.