

Evolutionary Many-Objective Optimization by NSGA-II and MOEA/D with Large Populations

Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto and Yusuke Nojima
Department of Computer Science and Intelligent Systems, Osaka Prefecture University
Gakuen-cho, Naka-ku, 1-1, Sakai, Osaka, 599-8534 Japan
{hisaoi@, sakane@ci., nori@ci., nojima@} cs.osakafu-u.ac.jp.

Abstract—Evolutionary multiobjective optimization (EMO) is an active research area in the field of evolutionary computation. EMO algorithms are designed to find a non-dominated solution set that approximates the entire Pareto front of a multiobjective optimization problem. Whereas EMO algorithms usually work well on two-objective and three-objective problems, their search ability is degraded by the increase in the number of objectives. One difficulty in the handling of many-objective problems is the exponential increase in the number of non-dominated solutions necessary for approximating the entire Pareto front. A simple countermeasure to this difficulty is to use large populations in EMO algorithms. In this paper, we examine the behavior of EMO algorithms with large populations (e.g., with 10,000 individuals) through computational experiments on multiobjective and many-objective knapsack problems with two, four, six, eight and ten objectives. We examine two totally different algorithms: NSGA-II and MOEA/D. NSGA-II is a Pareto dominance-based algorithm while MOEA/D uses scalarizing functions. Their search ability is examined for various specifications of the population size under the fixed computation load. That is, we use the total number of examined solutions as the stopping condition of each algorithm. Thus the use of a very large population leads to the termination at an early generation (e.g., 20th generation). It is demonstrated through computational experiments that the use of too large populations makes NSGA-II very slow and inefficient. On the other hand, MOEA/D works well even when it is executed with a very large population. We also discuss why MOEA/D works well even when the population size is unusually large.

Keywords—Evolutionary multiobjective optimization (EMO), evolutionary many-objective optimization, NSGA-II, MOEA/D, cellular genetic algorithms.

I. INTRODUCTION

Recently evolutionary multiobjective optimization (EMO) algorithms have been successfully used in various application areas [1]-[7]. EMO algorithms are usually designed to search for a non-dominated solution set that approximates the entire Pareto front of a multiobjective optimization problem. Well-known and frequently-used EMO algorithms such as SPEA [8] and NSGA-II [9] can be characterized by the use of Pareto dominance relation and a diversity maintenance mechanism for fitness evaluation together with some sort of elitism. Those algorithms are often called Pareto dominance-based algorithms because Pareto dominance relation is used as the primary fitness evaluation criterion. Pareto dominance-based algorithms usually work well on multiobjective problems with two or three objectives. However their search ability is often severely degraded by the increase in the number of objectives [10], [11].

The main difficulty of many-objective problems for Pareto dominance-based algorithms is that individuals with many objectives are not likely to be dominated by others. If all individuals in the current population are non-dominated, Pareto dominance-based fitness evaluation schemes cannot generate any selection pressure toward the Pareto front. In this case, fitness evaluation is based only on the secondary criterion: a diversity maintenance mechanism. As a result, individuals do not converge toward the Pareto front. In this manner, the convergence property of Pareto dominance-based algorithms is severely degraded by the increase in the number of objectives. This is the main difficulty in the handling of many-objective problems by Pareto dominance-based algorithms.

The increase in the number of objectives may exponentially increase the number of non-dominated solutions necessary for approximating the Pareto front of a multiobjective problem. In Fig. 1, we show 10 non-dominated solutions which seem to well approximate the Pareto front of a two-objective knapsack problem. We also show 40 non-dominated solutions of a three-objective knapsack problem. In Fig. 2, more non-dominated solutions seems to be needed to approximate the Pareto front whereas the non-dominated solution set in Fig. 1 looks a good approximation. As shown in Fig. 1 and Fig. 2, the Pareto front of a k -objective problem is usually a $(k-1)$ -dimensional hypersurface. Roughly and intuitively speaking, if we need 10 non-dominated solutions for approximating a tradeoff curve (i.e., a single-dimensional Pareto front for $k=2$), we may need 10^{k-1} non-dominated solutions for a $(k-1)$ -dimensional Pareto front.

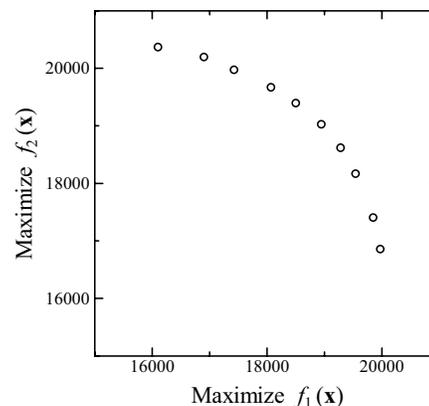


Figure 1. A set of 10 non-dominated solutions of a two-objective problem.

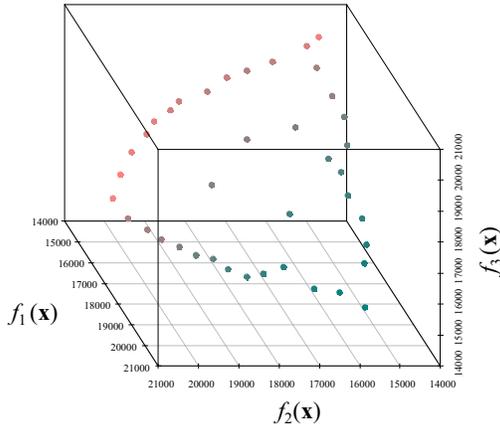


Figure 2. A set of 40 non-dominated solutions of a three-objective problem.

A simple and naive countermeasure against the exponential increase in the number of necessary non-dominated solutions is to use a large population with many individuals (e.g., 10,000 individuals). The use of such a large population may ease the deterioration in the convergence property of Pareto dominance-based algorithms. In this paper, we examine the behavior of EMO algorithms with large populations on many-objective problems. We use two totally different representative EMO algorithms: NSGA-II [9] and MOEA/D [12]. NSGA-II, which is based on Pareto dominance relation, is the most well-known and frequently-used EMO algorithm in the literature. On the other hand, MOEA/D is a recently developed EMO algorithm, which is not based on Pareto dominance relation. The main characteristic feature of MOEA/D is the use of scalarizing functions with uniformly distributed weight vectors (e.g., the weighted sum and the weighted Tchebycheff function). While the original version of MOEA/D has an archive population, we use its cellular version with no archive population [13]. This is because NSGA-II has no archive population (i.e., because we try to examine the behavior of these EMO algorithms under the same setting). Of course, it is an interesting future research issue to examine the behavior of EMO algorithms with archive populations under a large (or unlimited) upper bound on the archive population size.

This paper is organized as follows. First we briefly explain NSGA-II and MOEA/D in Section II. Next we explain why many-objective problems are difficult for Pareto dominance-based algorithms in Section III. Then we examine the behavior of NSGA-II and MOEA/D with large populations in Section IV. Finally we conclude this paper in Section V.

II. EVOLUTIONARY MULTI-OBJECTIVE OPTIMIZATION

A. Multiobjective Optimization Problems

A k -objective maximization problem can be written as

$$\text{Maximize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x})) \text{ subject to } \mathbf{x} \in \mathbf{X}, \quad (1)$$

where $\mathbf{f}(\mathbf{x})$ is the objective vector, $f_i(\mathbf{x})$ is the i -th objective function, \mathbf{x} is the decision vector, and \mathbf{X} is the feasible region.

Let \mathbf{y} and \mathbf{z} be two feasible solutions of the k -objective maximization problem in (1). If the following conditions hold, \mathbf{z} can be viewed as being better than \mathbf{y} :

$$\forall i: f_i(\mathbf{y}) \leq f_i(\mathbf{z}) \text{ and } \exists j: f_j(\mathbf{y}) < f_j(\mathbf{z}). \quad (2)$$

In this case, we say that \mathbf{z} dominates \mathbf{y} (equivalently \mathbf{y} is dominated by \mathbf{z} : \mathbf{z} is better than \mathbf{y}).

When \mathbf{y} is not dominated by any other feasible solutions, \mathbf{y} is referred to as a Pareto-optimal solution of a multiobjective optimization problem. The set of all Pareto-optimal solutions forms the tradeoff surface in the objective space. This tradeoff surface is referred to as the Pareto front. EMO algorithms are designed to search for a set of well-distributed non-dominated solutions that approximates the entire Pareto front very well.

B. NSGA-II

NSGA-II [9] is the most well-known and frequently-used EMO algorithm in the literature. Its fitness evaluation scheme is based on Pareto dominance relation. More specifically, the best rank (Rank 1) is assigned to all non-dominated individuals in the current population. Then all the Rank 1 individuals are tentatively removed from the current population. The next rank (Rank 2) is assigned to all the non-dominated individuals in the remaining population. This rank assignment procedure is iterated until ranks are assigned to all individuals in the current population. The assigned rank to each individual is used as the primary criterion in the binary tournament selection for parent selection. Individuals with the same rank are evaluated by the secondary criterion called the crowding distance. Roughly speaking, individual in less crowded regions in the objective space is viewed as being better than other individuals in more crowded regions if they have the same rank.

Let N_{pop} be the population size in NSGA-II. By iterating selection, crossover and mutation, N_{pop} offspring are generated. The generated offspring are added to the current population. The next population is constructed by choosing the best N_{pop} individuals from the enlarged current population of size $2N_{\text{pop}}$. Each individual in the enlarged population is evaluated by the rank assignment procedure and the crowding distance in the same manner as in the parent selection phase.

C. MOEA/D

Fitness evaluation in MOEA/D [12] is based on scalarizing functions with uniformly distributed weight vectors. As in [12], weight vectors are generated from the following equations:

$$w_1 + w_2 + \dots + w_k = 1, \quad (3)$$

$$w_i \in \left\{ 0, \frac{1}{H}, \frac{2}{H}, \dots, \frac{H}{H} \right\}, \quad i = 1, 2, \dots, k, \quad (4)$$

where H is a user-definable positive integer, which can be viewed as the granularity of weight value discretization. For example, we have 101 weight values 0, 0.01, ..., 1.00 in (4) when H is specified as $H=100$. In the case of $k=2$ (i.e., two objectives), this specification leads to 101 weight vectors (0, 1), (0.01, 0.99), ..., (1, 0).

MOEA/D is easily explained as a cellular EMO algorithm [13] with a neighborhood structure in the k -dimensional weight space. Whereas the original version of MOEA/D has an archive population, we use its cellular version with no archive. A single cell with a single individual is located at the same place as each weight vector in the k -dimensional weight space. That is, each cell has its own weight vector, which is used in the scalarizing function for evaluating the individual in that cell. In our computational experiments, we used the following weighted sum because it worked better than the weighted Tchebycheff function on many-objective 0/1 knapsack problems [15]:

$$f(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + \dots + w_k f_k(\mathbf{x}). \quad (5)$$

The main characteristic feature of MOEA/D is that each individual has a different weight vector. Thus the number of the weight vectors is the same as the population size. This means that the population size depends on the specification of H in (4) and the number of objectives k . Another characteristic feature of MOEA/D is the use of local selection. Neighbors of a cell are defined by the Euclidean distance between cells in the weight space. The number of neighbors of each cell (including itself) is prespecified in MOEA/D. When an offspring is to be generated for a cell, two parents are selected from its neighbors by random selection. Each neighbor is evaluated by the scalarizing function with the weight vector of the current cell. An offspring is generated by crossover and mutation, which is compared with the individual in the current cell using the scalarizing function. If the offspring is better, the current individual is replaced with the offspring. The offspring is also compared with each neighbor. The scalarizing function with the weight vector of each neighbor is used in the comparison. All neighbors, which are inferior to the offspring, are replaced with the offspring (i.e., local replacement).

III. WHY ARE MANY-OBJECTIVE PROBLEMS DIFFICULT?

In a k -dimensional unit hypercube $[0, 1]^k$, we randomly generated 200 vectors. Then we calculated the percentage of non-dominated vectors among them. The average percentage was calculated over ten runs for each k of $k=2, 4, \dots, 20$. Experimental results are shown in Fig. 3. From Fig. 3, we can see that almost all vectors are non-dominated when $k > 10$.

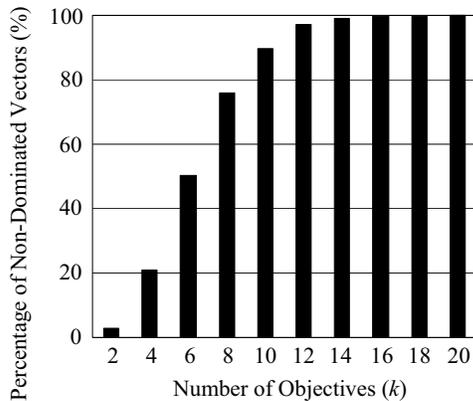


Figure 3. Percentage of non-dominated vectors among 200 random vectors.

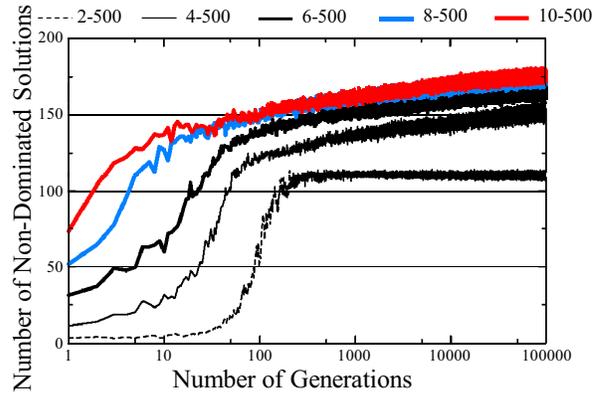


Figure 4. Number of non-dominated solutions before the generation update.

As in Sato et al. [16], we also monitored the number of non-dominated solutions during the execution of NSGA-II. NSGA-II was applied to multiobjective 500-item knapsack problems with two, four, six, eight and ten objectives. We denote a k -objective n -item knapsack problem as the k - n problem (i.e., 2-500, 4-500, 6-500, 8-500, and 10-500). We used the same 2-500 and 4-500 problems as in Zitzler & Thiele [8]. On the other hand, we generated our 6-500, 8-500 and 10-500 problems in the same manner as Zitzler & Thiele [8]. NSGA-II with the following specifications was applied to our test problems:

Population size: 100 individuals,
 Crossover probability: 0.8 (Uniform crossover),
 Mutation probability: 1/500 (Bit-flip mutation),
 Constraint handling: Greedy repair in Zitzler & Thiele [8],
 Number of runs: 10 runs.

In Fig. 4, we show the average number of non-dominated solutions over 10 runs in the enlarged population before the generation update at each generation. We can see from Fig. 4 that the increase in the number of non-dominated solutions was very fast in the case of many-objective problems. For example, the number of non-dominated solutions exceeded 100 after a few generations in the case of the 10-500 problem.

In the generation update phase of NSGA-II, the best 100 solutions are chosen from the enlarged population. If the enlarged population includes more than 100 non-dominated solutions, all the chosen 100 solutions are non-dominated. As a result, all solutions in the next generation are non-dominated. In this case, Pareto dominance relation has no effect on parent selection. That is, an individual with a larger crowding distance is always chosen as a parent in the binary tournament selection since all solutions have the same rank. In this manner, the selection pressure toward the Pareto front is severely weakened by the increase in the number of objectives.

IV. BEHAVIOR WITH LARGE POPULATIONS

We report experimental results by NSGA-II and MOEA/D with large populations on the 2-500, 4-500, 6-500, 8-500 and 10-500 problems. These two algorithms were compared under the same computation load. That is, we used the number of examined solutions as the stopping condition as shown in Table

I. For each test problem, we used three specifications of the population size in NSGA-II: 100 (small), 1000 (large), and 10000 (very large). We also examined three specifications for each test problem in MOEA/D as shown in Table II. It should be noted that the population size in MOEA/D is the same as the number of weight vectors, which is determined by the number of objectives and the granularity of weight values (i.e., k and H in Eq. (3)). In MOEA/D, the number of neighbors of each cell was specified as 5% of the population size in our computational experiments. For example, each cell has five neighbors (including itself) in the case of the population size 100.

TABLE I. STOPPING CONDITIONS (NUMBER OF EXAMINED SOLUTIONS)

Problem	2-500	4-500	6-500	8-500	10-500
Examined solutions	100,000	150,000	200,000	250,000	300,000

TABLE II. SPECIFICATIONS OF THE POPULATION SIZE IN MOEA/D

Problem	2-500	4-500	6-500	8-500	10-500
Size: Small	100	84	126	120	55
Size: Large	1000	969	792	792	715
Size: Very Large	10000	9880	8568	11440	11440

First we discuss the effect of increasing the population size on the computation time of each algorithm. Our computational experiments were performed on a PC with Intel(R) Xeon(R) CPU 5160 and 8.00 GB RAM using computer programs coded with Java. Experimental results are summarized in Table III and Table IV. The computation time of NSGA-II in Table III was drastically increased by the increase in the population size and the number of objectives. Such a drastic increase in the computation time makes it difficult to apply NSGA-II with very large populations to many-objective problems. On the other hand, the increase in the computation time is not so severe in Table IV. Even when MOEA/D was applied to the 10-500 problem with the very large population size 11440, the average computation time was 287.0 (sec.) whereas NSGA-II with similar conditions spent 2619.9 (sec.).

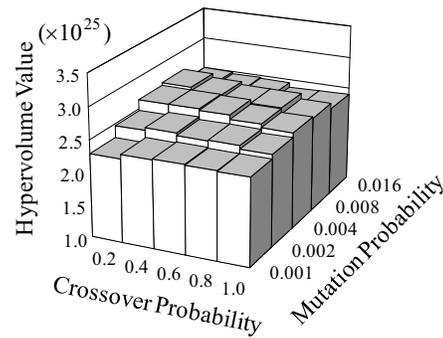
Using the hypervolume measure [17], we examined the effect of the population size on the search ability of each algorithm. One may think that the increase in the population size will improve the search ability of each algorithm whereas it increases the computation time. This may be almost always the case when computational experiments are performed under the fixed number of generations. It should be noted that our computational experiments were performed under the fixed number of examined solutions for each test problem as shown in Table I. Thus the increase in the population size means the decrease in the number of generations. In Fig. 5, we show experimental results by NSGA-II on the 6-500 problem. We examined five crossover probabilities (0.2, 0.4, 0.6, 0.8, 1.0) and five mutation probabilities (0.001, 0.002, 0.004, 0.008, 0.016) for each specification of the population size. From Fig. 5, we can see that the use of the very large population size (i.e., 10000) severely degraded the search ability of NSGA-II. Since the hypervolume calculation is time-consuming especially for many-objective problems, reported results in Fig. 5 are average results over five runs for each parameter specification.

TABLE III. AVERAGE COMPUTATION TIME OF NSGA-II (SECONDS)

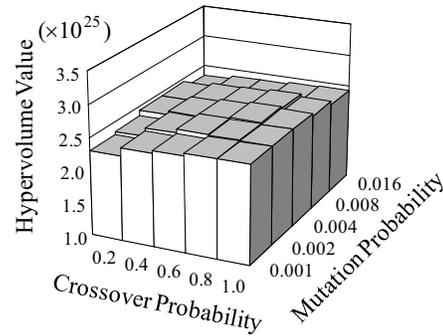
Problem	2-500	4-500	6-500	8-500	10-500
Size: 100	6.1	11.4	17.8	25.5	34.4
Size: 1000	24.7	44.5	73.7	111.6	152.4
Size: 10000	619.2	744.5	1079.5	1939.3	2619.9

TABLE IV. AVERAGE COMPUTATION TIME OF MOEA/D (SECONDS)

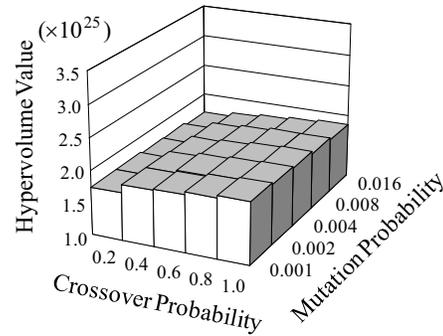
Problem	2-500	4-500	6-500	8-500	10-500
Size: Small	4.1	6.8	10.0	13.4	17.2
Size: Large	6.0	8.6	11.6	15.6	19.9
Size: Very Large	264.4	202.9	150.2	283.3	287.0



(a) Population size: 100.

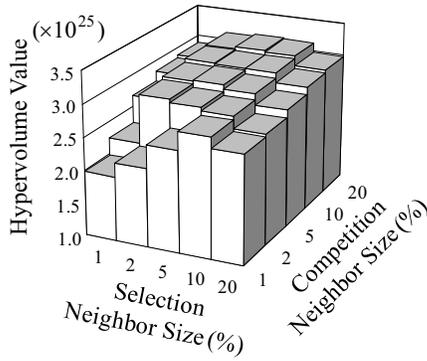


(b) Population size: 1000.

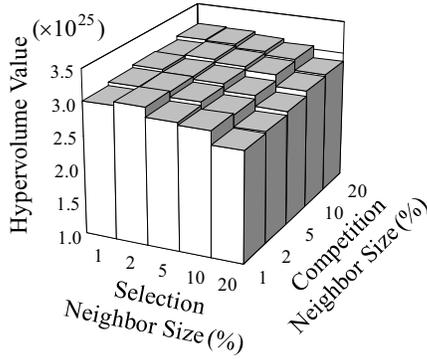


(c) Population size: 10000.

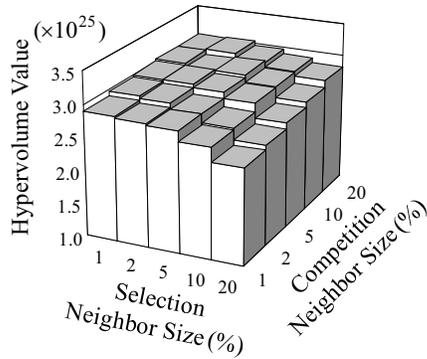
Figure 5. Experimental results by NSGA-II on the 6-500 problem.



(a) Population size: 126.



(b) Population size: 792.



(c) Population size: 8568.

Figure 6. Experimental results by MOEA/D on the 6-500 problem.

In Fig. 6, we show experimental results by MOEA/D on the 6-500 problem. We examined five specifications of the number of neighbors for parent selection and solution replacement: 1%, 2%, 5%, 10% and 20%. We examined all the 25 combinations of the two neighborhood structures. Average results were calculated over five runs for each combination. The crossover and mutation probabilities were specified as 0.8 and 0.002, respectively. In Fig. 6, the performance of MOEA/D was not degraded even when the population size was very large (8568).

Using the non-dominated solution sets obtained by each algorithm in our previous computational experiments in Table

III and Table IV, we calculated the average number of obtained non-dominated solutions over 30 runs for each experiment setting. When multiple solutions were located at the same point in the objective space, we counted them as a single solution. That is, overlapping solutions in the objective space were viewed as the same solution when we counted the number of non-dominated solutions. Average experimental results are summarized in Table V and Table VI.

TABLE V. AVERAGE NUMBER OF OBTAINED NON-DOMINATED SOLUTIONS BY NSGA-II IN THE COMPUTATIONAL EXPERIMENTS IN TABLE III

Problem	2-500	4-500	6-500	8-500	10-500
Size: 100	56.5	97.2	97.8	98.3	98.7
Size: 1000	53.6	889.5	961.0	974.4	978.2
Size: 10000	8.4	162.3	830.7	4575.1	8152.9

TABLE VI. AVERAGE NUMBER OF OBTAINED NON-DOMINATED SOLUTIONS BY MOEA/D IN THE COMPUTATIONAL EXPERIMENTS IN TABLE IV

Problem	2-500	4-500	6-500	8-500	10-500
Size: Small	9.7	33.4	59.1	75.0	50.7
Size: Large	11.3	73.8	133.6	178.4	224.4
Size: Very Large	22.6	92.9	216.8	380.3	553.7

From the comparison between Table V and Table VI, we can see that more non-dominated solutions were obtained by NSGA-II in Table V than MOEA/D in Table VI in almost all cases (especially for many-objective problems such as 8-500 and 10-500). This is because the crowding distance has a much larger effect than Pareto dominance on the fitness evaluation in NSGA-II in its application to many-objective problems. That is, the diversity maintenance mechanism played a much larger role than the selection pressure toward the Pareto front in the evolution by NSGA-II for many-objective problems.

In MOEA/D, multiple neighbors can be replaced with a good offspring. This means that neighboring cells are likely to have the same individual. This may explain why the number of obtained non-dominated solutions was much smaller than the population size in some cases in Table VI. In Table VII, we show experimental results by MOEA/D where the size of the competition neighborhood was specified as one. In this specification, no neighbors can be replaced with any offspring generated for a different cell. The current solution in each cell can be replaced only with the new offspring generated for the same cell. The size of the selection neighborhood was specified as 5% of the population size in Table VII as in Table VI. It is clear from the comparison between Table VI and Table VII that the number of obtained non-dominated solutions was increased by decreasing the size of the competition neighborhood from 5% of the population size to one in MOEA/D.

TABLE VII. AVERAGE NUMBER OF OBTAINED NON-DOMINATED SOLUTIONS BY MOEA/D WHERE THE COMPETITION NEIGHBORHOOD SIZE WAS ONE.

Problem	2-500	4-500	6-500	8-500	10-500
Size: Small	13.7	46.7	72.3	85.5	52.8
Size: Large	69.4	339.0	303.9	351.4	309.2
Size: Very Large	8.2	156.0	635.3	2158.3	3193.7

Whereas the use of the small competition neighborhood can increase the number of obtained non-dominated solutions (i.e., increase the diversity of solutions), it severely degrades the search ability of MOEA/D especially when the population size is large. In Fig. 7, we show experimental results of MOEA/D with the population size 8568 and small neighborhood on the 6-500 problem. It should be noted that the size of neighborhood in Fig. 7 is not specified by the percentage of the population size but the number of neighbors.

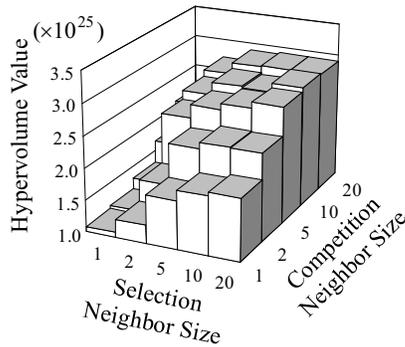


Figure 7. Experimental results by MOEA/D with the population size 8568 on the 6-500 problem. Much smaller neighborhood was used in this figure than Fig. 6 (c).

V. CONCLUDING REMARKS

We demonstrated that the search ability of MOEA/D was not degraded by the use of very large populations (i.e., a population with 10000 individuals) whereas NSGA-II did not work well with very large populations. The efficiency of MOEA/D was not degraded by the increase in the population size because multiple neighbors can be replaced with a newly generated good offspring. We also demonstrated that the computation time of MOEA/D was not severely increased by the use of large populations whereas the execution of NSGA-II was severely slowed down by the increase of the population size. This is because the fitness evaluation in MOEA/D is based on scalarizing functions whereas NSGA-II uses Pareto dominance relation. Even when the population size was small (as well as the case of large populations), better results were obtained by MOEA/D on the 6-objective 500-item knapsack problems than NSGA-II.

Our experimental results suggest that we can use MOEA/D to search for a large number of non-dominated solutions of many-objective problems. A large number of obtained non-dominated solutions by MOEA/D are likely to approximate the entire Pareto front better than those by Pareto dominance-based algorithms. One may think that the decision maker needs only a small number of non-dominated solutions. This may be true in many cases in real-world applications. Even in those cases, a large number of obtained non-dominated solutions can be utilized as candidate solutions from which only a small number of representative solutions are selected to be presented to the decision maker [18]. A large number of obtained non-dominated solutions by MOEA/D can be also used to support efficient interaction with the decision maker in interactive EMO algorithms. Since we have already had a large number of non-dominated solutions, no additional computation time is

needed to search for new candidate solutions to be presented to the decision maker.

ACKNOWLEDGMENT

This work was supported in part by the Japan Society for the Promotion of Science under Grant-in-Aid for Scientific Research (B) (20300084).

REFERENCES

- [1] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, John Wiley & Sons, Chichester, 2001.
- [2] C. A. C. Coello, D. A. van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer, Boston, 2002.
- [3] C. A. C. Coello and G. B. Lamont, *Applications of Multi-Objective Evolutionary Algorithms*, World Scientific, Singapore, 2004.
- [4] A. Abraham, L. C. Jain, and R. Goldberg (eds.), *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*, Springer, Berlin, 2005.
- [5] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications*, Springer, Berlin, 2005.
- [6] Y. Jin (ed.), *Multi-Objective Machine Learning*, Springer, Berlin, 2006.
- [7] A. Ghosh, S. Dehuri, and S. Ghosh (Eds.), *Multi-Objective Evolutionary Algorithms for Knowledge Discovery from Data Bases*, Springer, Berlin, 2009.
- [8] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. on Evolutionary Computation* 3 (4), pp. 257-271, November 1999.
- [9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. on Evolutionary Computation* 6 (2), pp. 182-197, April 2002.
- [10] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," *Proc. of 2008 IEEE Congress on Evolutionary Computation*, pp. 2424-2431, 2008.
- [11] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima, "Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems," *Proc. of 2008 Genetic and Evolutionary Computation Conference*, pp. 649-656, 2008.
- [12] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. on Evolutionary Computation* 11 (6), pp. 712-731, December 2007.
- [13] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization," *Proc. of 2009 IEEE Congress on Evolutionary Computation*, pp. 2508-2515, 2009.
- [14] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithm," *Lecture Notes in Computer Science 1993: Evolutionary Multi-Criterion Optimization - EMO 2001*, pp. 82-95, Springer, Berlin, March 2001.
- [15] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," *Proc. of 5th International Conference on Evolutionary Multi-Criterion Optimization*, pp. 438-452, 2009.
- [16] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," *Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007*, pp. 5-20, Springer, Berlin, March 2007.
- [17] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms - A comparative case study," *Lecture Notes in Computer Science 1498: Parallel Problem Solving from Nature - PPSN V*, pp. 292-301, Springer, Berlin, September 1998.
- [18] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Single-objective and multi-objective formulations of solution selection for hypervolume maximization," *Proc. of 2009 Genetic and Evolutionary Computation Conference* (in press).