H. Ishibuchi, Y. Sakane, N. Tsukamoto and Y. Nojima, "Simultaneous use of different scalarizing functions in MOEA/D," *Proc. of Genetic and Evolutionary Computation Conference - GECCO 2010*, pp. 519-526, Portland, USA, July 7-11, 2010.

Simultaneous Use of Different Scalarizing Functions in MOEA/D

Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto and Yusuke Nojima
Graduate School of Engineering, Osaka Prefecture University
1-1 Gakuen-cho, Naka-ku, Sakai, Osaka 599-8531, Japan
Phone: +81-72-254-9350
{hisaoi@, sakane@ci, nori@ci., nojima@}cs.osakafu-u.ac.jp

ABSTRACT
The use of Pareto dominance for fitness evaluation has been the mainstream in evolutionary multiobjective optimization for the last two decades. Recently, it has been pointed out in some studies that Pareto dominance-based algorithms do not always work well on multiobjective problems with many objectives. Scalarizing function-based fitness evaluation is a promising alternative to Pareto dominance especially for the case of many objectives. A representative scalarizing function-based algorithm is MOEA/D (multiobjective evolutionary algorithm based on decomposition) of Zhang & Li (2007). Its high search ability has already been shown for various problems. One important implementation issue of MOEA/D is a choice of a scalarizing function because its search ability strongly depends on this choice. It is, however, not easy to choose an appropriate scalarizing function for each multiobjective problem. In this paper, we propose an idea of using different types of scalarizing functions simultaneously. For example, both the weighted Tchebycheff (Chebyshev) and the weighted sum are used for fitness evaluation. We examine two methods for implementing our idea. One is to use multiple grids of weight vectors and the other is to assign a different scalarizing function alternately to each weight vector in a single grid.

Categories and Subject Descriptors
I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search – *Heuristic Methods.*

General Terms
Algorithms.

Keywords
Evolutionary multiobjective optimization (EMO), scalarizing function, MOEA/D.

1. INTRODUCTION
Since Goldberg's suggestion in the late-80s [10], the use of Pareto dominance for fitness evaluation has been the mainstream in the field of evolutionary multiobjective optimization (EMO) for the last two decades [4]-[6], [26]. Almost all well-known and frequently-used EMO algorithms such as NSGA-II [7] and SPEA [31] are based on Pareto dominance. Such a Pareto dominance-based algorithm usually works well on multiobjective problems with two or three objectives. Its search ability is, however, often severely degraded by the increase in the number of objectives as pointed out in the literature [18]-[21], [25], [32]. This is because almost all individuals in a population become non-dominated with each other under many objectives [13]. When all individuals in a population are non-dominated, Pareto dominance-based fitness evaluation cannot generate any selection pressure toward the Pareto front.

A theoretically well-supported alternative to Pareto dominance is the use of an indicator function such as the hypervolume measure [1], [2], [9], [27], [29], [30]. A class of EMO algorithms with indicator-based fitness evaluation is referred to as IBEAs (indicator-based evolutionary algorithms). High search ability of IBEAs has been demonstrated in the literature [27]. One practical difficulty of IBEAs in their applications to many-objective problems is that the computation time for the hypervolume calculation exponentially increases with the number of objectives.

Another alternative to Pareto dominance-based fitness evaluation is the use of scalarizing functions. It has been demonstrated in the literature [8], [11], [12], [14], [15], [19], [20] that better results can be obtained by scalarizing function-based algorithms than Pareto dominance-based ones for combinatorial or many-objective problems with more than three objectives. The main advantage of scalarizing function-based algorithms over other EMO algorithms is the simplicity of fitness evaluation. Scalarizing functions can be easily calculated even when we have many objectives.

A well-known representative EMO algorithm with scalarizing function-based fitness evaluation is MOEA/D (multiobjective evolutionary algorithm based on decomposition) proposed by Zhang & Li [28]. This is a simple but powerful EMO algorithm. It has been reported in the literature [3], [17], [22], [24] that MOEA/D works very well on a wide range of multiobjective problems with many objectives, discrete decision variables and/or complicated Pareto sets.

One important implementation issue of MOEA/D is a choice of an appropriate scalarizing function for a particular multiobjective problem at hand. In the original version of MOEA/D [28], the weighted sum and the weighted Tchebycheff (Chebyshev) were examined. The performance of MOEA/D strongly depends on the choice of a scalarizing function [16]. Whereas an idea of automatically switching between the weighted sum and the weighted Tchebycheff was proposed [16], good results were not reported due to the difficulty in designing an effective mechanism for choosing an appropriate scalarizing function between them.

In this paper, we propose an idea of simultaneously using different types of scalarizing functions in MOEA/D in order to alleviate the difficulty in choosing an appropriate scalarizing function for each multiobjective problem. For example, we use both the weighted sum and the weighted Tchebycheff instead of choosing one of them. Two implementation schemes of the proposed idea are examined in this paper. One is to use multiple grids of weight vectors where each grid is used by a single scalarizing function. The other is to use different types of scalarizing functions in a single grid of weight vectors where a different scalarizing function is alternately assigned to each weight vector.

For example, let us assume that the weighted sum and the weighted Tchebycheff with a set of six weight vectors (1.0, 0.0), (0.8, 0.2), ..., (0.0, 1.0) are used in MOEA/D. The set of these six weight vectors can be viewed as a grid with the six points in the two-dimensional weight vector space. Two grids with the six weight vectors are used in the multi-grid implementation scheme. One is for the weighted sum, and the other is for the weighted Tchebycheff. As a result, MOEA/D has six weighted sum functions and six weighted Tchebycheff functions. On the other hand, only a single grid is used in the single-grid implementation scheme. The weighted sum and the weighted Tchebycheff are alternately assigned to each weight vector in the grid (e.g., the weighted sum is assigned to (1.0, 0.0), (0.6, 0.4) and (0.2, 0.8) while the weighted Tchebycheff is assigned to (0.8, 0.2), (0.4, 0.6) and (0.0, 1.0)). As a result, MOEA/D has three weighted sum functions and three weighted Tchebycheff functions.

This paper is organized as follows. First we briefly explain MOEA/D as a cellular EMO algorithm with a grid in the weight vector space in Section 2. Next we demonstrate the sensitivity of the performance of MOEA/D to the choice of a scalarizing function in Section 3. Then we explain our idea (i.e., the simultaneous use of different types of scalarizing functions in MOEA/D) and its two implementation schemes in Section 4. The effectiveness of our idea is examined in Section 5. Finally we conclude this paper in Section 6.

2. MOEA/D ALGORITHMS

2.1 Scalarizing Functions

An m-objective maximization problem can be written as

$$\text{Maximize } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ ...,\ f_m(\mathbf{x}))\,, \tag{1}$$

where $\mathbf{f}(\mathbf{x})$ is an m-dimensional objective vector, $f_i(\mathbf{x})$ is the i-th objective to be maximized, and $\mathbf{x}$ is a decision vector.

One of well-known and frequently-used scalarizing functions is the weighted sum. The weighted sum with a non-negative weight vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_m)$ is written as

$$g^{WS}(\mathbf{x}\,|\,\boldsymbol{\lambda}) = \lambda_1 \cdot f_1(\mathbf{x}) + \lambda_2 \cdot f_2(\mathbf{x}) + \ ... \ + \lambda_m \cdot f_m(\mathbf{x})\,, \tag{2}$$

where λ_i is a non-negative weight for the i-th objective $f_i(\mathbf{x})$. We assume that the weight vector $\boldsymbol{\lambda}$ satisfies $\lambda_1 + \lambda_2 + ... + \lambda_m = 1$ and $\lambda \geq 0$ for $i = 1, 2, ..., m$. The weighted sum in (2) is maximized.

Another well-known and frequently-used scalarizing function is the weighted Tchebycheff. The weighed Tchebycheff with a reference point $\mathbf{z}^* = (z_1^*, z_2^*, ..., z_m^*)$ in the objective space and the weight vector $\boldsymbol{\lambda} = (\lambda_1, \lambda_2, ..., \lambda_m)$ is written as

$$g^{TE}(\mathbf{x}\,|\,\boldsymbol{\lambda}, \mathbf{z}^*) = \max_{i=1,2,...,m} \{\lambda_i \cdot |\, z_i^* - f_i(\mathbf{x})\,|\}\,. \tag{3}$$

In the same manner as in Zhang & Li [28], we update the reference point $\mathbf{z}^*$ during the execution of MOEA/D as

$$z_i^* = 1.1 \cdot \max\{f_i(\mathbf{x})\,|\,\mathbf{x} \in \Omega\},\ i = 1, 2, ..., m\,, \tag{4}$$

where Ω shows all the examined solutions during the execution of MOEA/D. The reference point is updated when the maximum value of each objective in (4) is updated. The weighted Tchebycheff in (3) is minimized to maximize each objective.

The weighted sum and the weighed Tchebycheff were used in MOEA/D of Zhang & Li [28]. In this paper, we also examine the following augmented weighted Tchebycheff for comparison:

$$g^{AT}(\mathbf{x}\,|\,\boldsymbol{\lambda}, \mathbf{z}^*) = \max_{i=1,2,...,m} \{\lambda_i \cdot |\, z_i^* - f_i(\mathbf{x})\,|\} + \rho \sum_{j=1}^{m} |\, f_j(\mathbf{x}) - z_j^*\,|\,, \tag{5}$$

where ρ is usually a very small positive constant (e.g., 0.1).

2.2 Our Cellular Implementation of MOEA/D

MOEA/D of Zhang & Li [28] is a simple but powerful scalarizing function-based algorithm. MOEA/D has a number of advantages over Pareto dominance-based algorithms such as the scalability to many-objective problems, high search ability for combinatorial optimization, computational efficiency of fitness evaluation, and high compatibility with local search.

The main characteristic feature of MOEA/D is the handling of a multiobjective problem as a collection of a large number of single-objective problems. Each single-objective problem has a scalarizing function with a different weight vector. Each weight vector has a single individual in the current population. This idea is similar to a cellular EMO algorithm of Murata et al. [23] where a different weight vector was assigned to each cell. In both algorithms, each individual in the current population was governed by a scalarizing function with a different weight vector.

MOEA/D uses a pre-specified number of uniformly distributed weight vectors satisfying the following two conditions:

$$\lambda_1 + \lambda_2 + ... + \lambda_m = 1\,, \tag{6}$$

$$\lambda_i \in \left\{0, \frac{1}{H}, \frac{2}{H}, ..., \frac{H}{H}\right\},\ i = 1, 2, ..., m\,, \tag{7}$$

where H is a user-definable positive integer. The number of weight vectors is calculated as $N = {}_{H+m-1}C_{m-1}$ [28]. For example, we have 101 weight vectors by specifying H as $H = 100$ for a two-objective problem: $\boldsymbol{\lambda}$ = (0, 1), (0.01, 0.99), (0.02, 0.98), ..., (1, 0). In Fig. 1, we show 15 weight vectors for the case of $m = 3$ and $H = 4$. Fig. 1 shows that a set of weight vectors satisfying (6) and (7) can be viewed as a $(m-1)$-dimensional grid in the m-dimensional weight vector space $[0, 1]^m$.

Let us denote the generated N weight vectors as $\{\boldsymbol{\lambda}^1, \boldsymbol{\lambda}^2, ..., \boldsymbol{\lambda}^N\}$. Each weight vector $\boldsymbol{\lambda}^k$ has the nearest T weight vectors (including

λ^k itself) as its neighbors where T is a user-definable positive integer. We denote the T neighbors of λ^k by $B(\lambda^k)$, which can be viewed as the neighborhood of size T for the weight vector λ^k. The distance between two weight vectors is measured by the standard Euclidean distance.

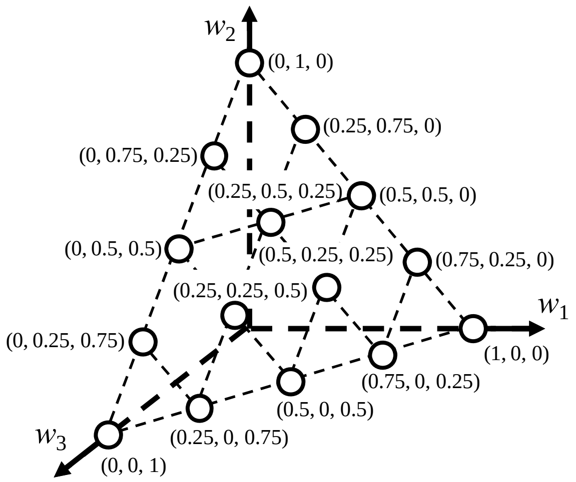


Figure 1. Weight vectors for three-objective problems (H = 4).

The same neighborhood structure is also used for individuals since each weight vector has a single individual. Let us denote the individual associated with the weight vector λ^k by $\mathbf{x}^k$. Then we denote the T individuals associated with the T weight vectors in $B(\lambda^k)$ by $B(\mathbf{x}^k)$, which is referred to as the neighborhood of $\mathbf{x}^k$. We also call the T individuals in $B(\mathbf{x}^k)$ as the neighbors of $\mathbf{x}^k$. In MOEA/D, genetic operations for each individual are locally performed among its neighbors as in cellular algorithms.

Let us assume that we have N weight vectors. We also have the T neighbors in $B(\lambda^k)$ for each weight vector λ^k, $k = 1, 2, ..., N$. As in standard evolutionary algorithms, the first step of MOEA/D is to generate an initial population. It should be noted that the population size is the same as the number of the weight vectors (i.e., N). We first randomly generate an initial individual for each weight vector. Next we generate an offspring for each weight vector by selection, crossover and mutation. When an offspring is to be generated for the weight vector λ^k, a couple of parents are randomly selected among the T neighbors of $\mathbf{x}^k$ in $B(\mathbf{x}^k)$. Then an offspring is generated by crossover and mutation. Let us denote the generated offspring by $\mathbf{y}^k$. If the offspring $\boldsymbol{y}^k$ is better than the current individual $\mathbf{x}^k$, $\mathbf{x}^k$ is replaced with $\boldsymbol{y}^k$. The two individuals $\mathbf{x}^k$ and $\mathbf{y}^k$ are compared with each other by the scalarizing function with the weight vector λ^k (i.e., the weighted sum or the weighted Tchebycheff in MOEA/D of Zhang & Li [28]). The newly generated offspring $\mathbf{y}^k$ is also compared with all neighbors in $B(\mathbf{x}^k)$. This comparison is performed using the weight vector of each neighbor. If $\mathbf{y}^k$ is better than some neighbors, they are replaced with $\mathbf{y}^k$. The genetic operations (i.e., selection, crossover, mutation) and the comparison of the newly generated offspring with all the T neighbors in $B(\mathbf{x}^k)$ are performed for each individual $\mathbf{x}^k$ (i.e., $k = 1, 2, ..., N$) in the current population. We used the total number of examined solutions as the stopping condition in our computational experiments in this paper.

As we have already explained, MOEA/D is based on local selection and local replacement. Local selection means the choice of parents from the neighbors of the current solution while local replacement means the comparison of the newly generated offspring with its neighbors for replacement. In the original MOEA/D of Zhang & Li [28], the same neighborhood structure was used for local selection and local replacement. In this paper, we examine different specifications of the number of neighbors for local selection and local replacement.

The original version of MOEA/D [28] has a secondary population for combinatorial optimization (whereas MOEA/D does not have it for continuous optimization). No individual in the secondary population is used in the genetic operations for generating new offspring. This means that the secondary population has no effect on the search behavior of MOEA/D. In MOEA/D, the "replace-if-better" strategy is used for all individuals as in cellular algorithms. This replacement strategy can be viewed as a kind of elitism. Thus MOEA/D has high search ability without utilizing non-dominated solutions in the secondary population as parents in the genetic operations for generating new offspring.

The size of the secondary population often becomes very large especially in the case of many objectives. This is because almost all solutions are non-dominated with each other under many objectives. As a result, the maintenance of the secondary population often needs a long computational time when we have many objectives. In our implementation of MOEA/D in this paper, we do not use any secondary population in order to avoid the severe increase in the computation time.

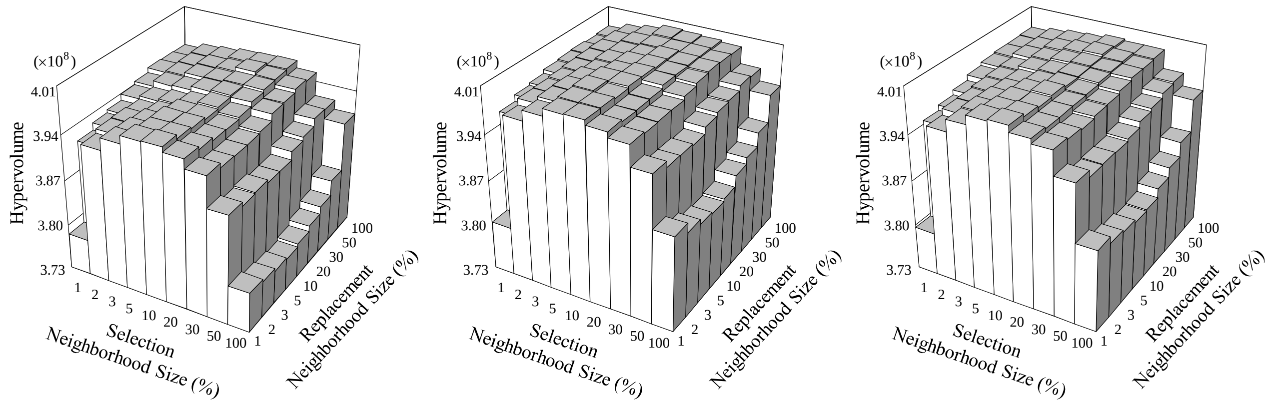
3. EXPERIMENTAL RESULTS

In this section, we demonstrate the sensitivity of the performance of MOEA/D on the choice of a scalarizing function through computational experiments on 500-item 0/1 knapsack problems with two and four objectives of Zitzler & Thiele [31]. We also generated a 500-item 0/1 knapsack problem with six objectives in the same manner as in [31]. In this paper, we denote the m-objective n-item 0/1 knapsack problem as the m-n problem (i.e., 2-500, 4-500 and 6-500).

In our computational experiments, solutions of each test problem were coded as binary strings of length 500. We applied the same greedy repair method as in Zitzler & Thiele [31] to infeasible solutions. The following setting was used in MOEA/D:

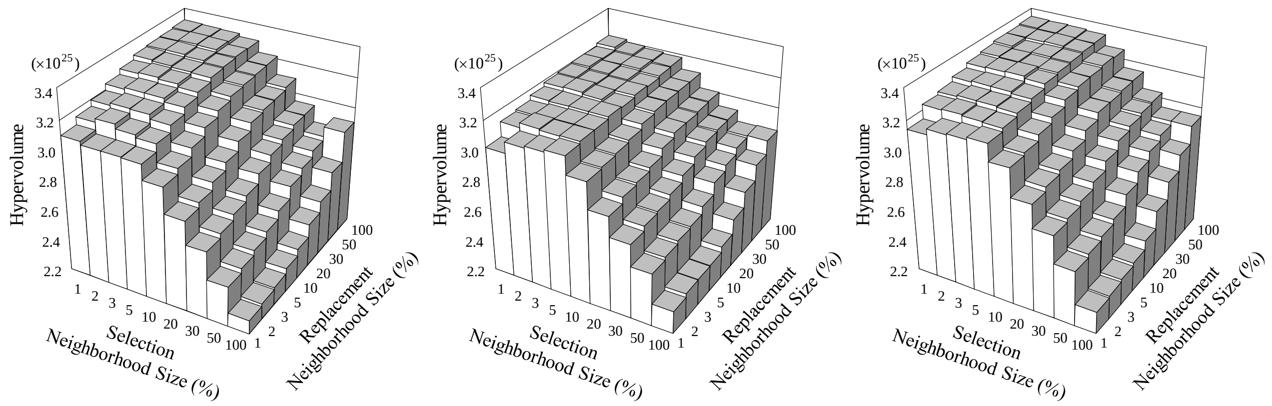
Population size: 200 ($H = 199$) for 2-500, 220 ($H = 9$) for 4-500,
252 ($H = 5$) for 6-500.
Crossover probability: 0.9 (Uniform crossover).
Mutation probability: 0.004 (Bit-flip mutation).
Selection neighborhood size (Percentage of the population):
1%, 2%, 3%, 5%, 10%, 20%, 30%, 50%, 100%.
Replacement neighborhood size (Percentage of the population):
1%, 2%, 3%, 5%, 10%, 20%, 30%, 50%, 100%.
Stopping condition: 200,000 solution evaluations.

We examined all the 9×9 combinations of the nine specifications of the neighborhood size for local selection and local replacement. For each combination, the average value of the hypervolume measure was calculated over 100 runs for each test problem. This computational experiment was performed using each of the three scalarizing functions. Experimental results on the 2-500 and 6-500 test problems are summarized in Fig. 2 and Fig. 3, respectively.



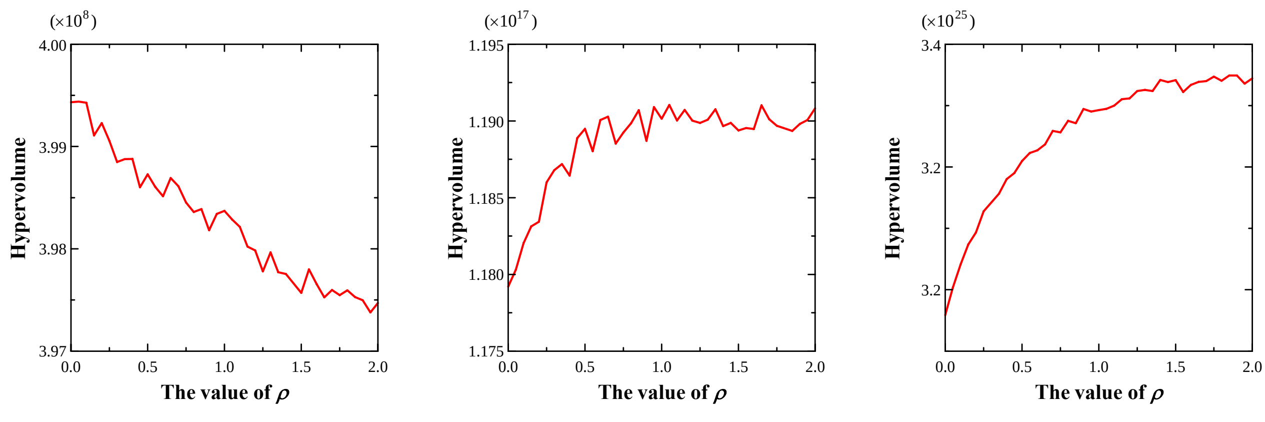
(a) Weighted sum. (b) Weighted Tchebycheff. (c) Augmented Tchebycheff ($\rho = 0.75$).

Figure 2. Experimental results on the 2-500 knapsack problem.



(a) Weighted sum. (b) Weighted Tchebycheff. (c) Augmented Tchebycheff ($\rho = 0.75$).

Figure 3. Experimental results on the 6-500 knapsack problem.



(a) 2-500 knapsack problem. (b) 4-500 knapsack problem. (c) 6-500 knapsack problem.

Figure 4. Effect of the value of ρ on the performance of the augmented weighted Tchebycheff.

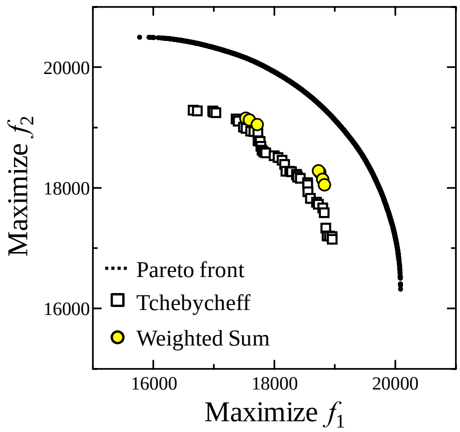
The weighted Tchebycheff works well on the 2-500 problem in Fig. 2 whereas the weighted sum works well on the 6-500 problem in Fig. 3. The augmented weighted Tchebycheff works well on both the 2-500 and 6-500 problems. The augmented weighted Tchebycheff, however, has an additional parameter ρ. Very small value of ρ is often used in the literature. We examined

41 different specifications of ρ (ρ=0.00, 0.05, ..., 2.00) in Fig. 4 where the size of the selection neighborhood and the replacement neighborhood was specified as 3% and 30% of the population size. Fig. 4 shows average results over 100 runs. As shown in Fig. 4, an appropriate specification of ρ is problem-dependent.

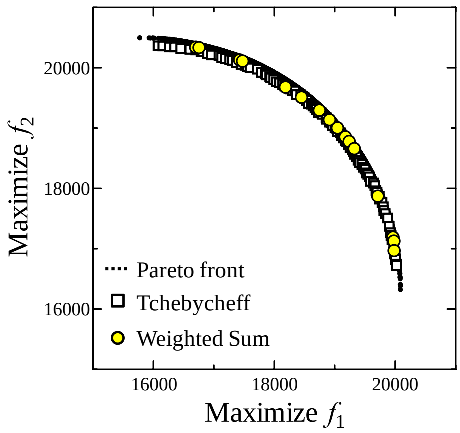
We can see from Figs. 2-4 that the performance of MOEA/D depends on the choice of a scalarizing function. These figures also show that an appropriate choice is problem-dependent.

The weighted sum and the weighted Tchebycheff are compared with each other in Fig. 5 where experimental results of a single run of MOEA/D with each function are depicted. The selection neighborhood and the replacement neighborhood were specified as 3% and 30% of the population size in Fig. 5, respectively.

Fig. 5 shows that the weighted sum has higher convergence ability to drive the population toward the Pareto front whereas it does not have high diversity maintenance ability to widen the population along the Pareto front. On the other hand, the weighted Tchebycheff has higher diversity maintenance ability whereas its convergence ability seems to be inferior to the weighted sum.



(a) At the 50th generation.



(b) At the 5000th generation.

Figure 5. Experimental results of a single run of MOEA/D with the weighted sum and the weighted Tchebycheff on the 2-500 problem.

4. THE PROPOSED IDEA

As shown in Figs. 2-5, each scalarizing function has its own advantages and disadvantages. The choice of an appropriate scalarizing function is problem-dependent. Moreover, the choice of an appropriate parameter value for ρ in the augmented weighted Tchebycheff is also problem-dependent. Our idea is to utilize the advantages of each scalarizing function in a single MOEA/D algorithm. That is, our idea is to simultaneously use multiple scalarizing functions in a single MOEA/D algorithm. In particular, we examine the simultaneous use of the weighted sum and the weighted Tchebycheff in this paper. Our idea, however, is applicable to other scalarizing functions (i.e., the simultaneous use of the three scalarizing functions in Section 2).

We propose the following two implementation schemes of our idea. One is a multi-grid scheme where each scalarizing function has its own complete grid of weight vectors. This implementation is illustrated in Fig. 6 where the weighted sum and the weighted Tchebycheff are used in a single MOEA/D algorithm. In Fig. 6, each scalarizing function has its own complete grid with 15 weight vectors (Such as small grid is used only for illustration purposes). The two grids with the same 15 weight vectors are simultaneously used in MOEA/D. These two grids can be viewed as overlapping with each other as shown in Fig. 6. As a result, the population size becomes 30. When we use two grids of weight vectors as in Fig. 6, the number of neighbors also becomes twice from the case of a single-grid. For example, let us assume that the number of neighbors is three (i.e., $T=3$) in Fig. 6 in the original MOEA/D algorithm. In this case, the three neighbors of the weight vector (1, 0, 0) are (1, 0, 0), (0.75, 0.25, 0) and (0.75, 0, 0.25). When we use two grids as in Fig. 6, the same three weight vectors are the neighbors of (1, 0, 0). However, there exist two overlapping weight vectors at each location of weight vectors. So the number of neighbors is actually six in Fig. 6 when $T=3$.

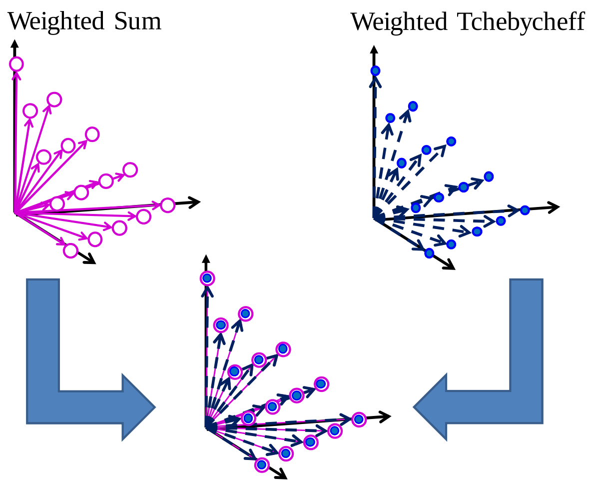


Figure 6. Multi-grid implementation scheme.

The other implementation is to assign a different scalarizing function alternately to each weight vector in a single grid. This implementation scheme is illustrated in Fig. 7 where the weighted sum and the weighted Tchebycheff are used. As shown in Fig. 7, each scalarizing function can be viewed as having an incomplete grid of weight vectors. MOEA/D with this implementation has a single complete grid with 15 weight vectors as in the original MOEA/D algorithm [28]. However, each weight vector has a different scalarizing function in our single-grid implementation scheme with multiple scalarizing functions. It is easy to generalize the single-grid implementation in Fig. 7 with two scalarizing functions to the case with more than two functions.

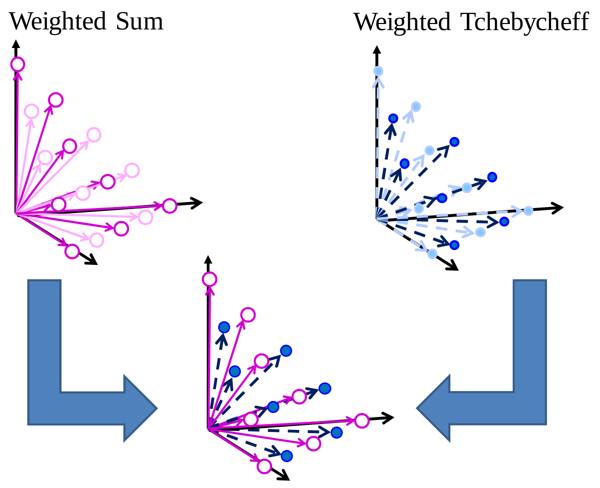


Figure 7. Single-grid implementation scheme.

5. EFFECTS OF THE PROPOSED IDEA

In this section, we examine how the simultaneous use of multiple scalarizing functions can improve the performance of MOEA/D with a single scalarizing function. We also compare the two implementation schemes of the proposed idea with each other.

In our computational experiments in the section, we used the following five variants of MOEA/D:

Weighted Sum: MOEA/D with the weighted sum in (2)
Tchebycheff: MOEA/D with the weighted Tchebycheff in (3)
Multi-Grid: MOEA/D with the multi-grid scheme in Fig. 6
Single-Grid: MOEA/D with the single-grid scheme in Fig. 7
Augmented: MOEA/D with the augmented Tchebycheff in (5)

Each variant was applied to the 2-500, 4-500 and 6-500 problems using the same conditions as in Section 3. The neighborhood size was specified as follows: the selection neighborhood and the replacement neighborhood was 3% and 30% of the population, respectively. These specifications of the neighborhood size are based on the experimental results in Figs. 2-4 (i.e., since good results were obtained in Figs. 2-4 from these specifications).

Experimental results are summarized in Figs. 8-10 where the distribution of the hypervolume values over 100 runs of each algorithm on each test problem is depicted as a histogram. In Fig. 8, the best results were obtained by the weighted Tchebycheff, and the multi-grid and single-grid schemes for the 2-500 problem. On the other hand, it is clear that the simultaneous use of the weighted sum and the weighted Tchebycheff (i.e., the multi-grid and single-grid schemes) outperformed their individual use in Fig. 9 for the 4-500 test problem and Fig. 10 for the 6-500 test problem. This observation shows the effectiveness of the proposed idea.

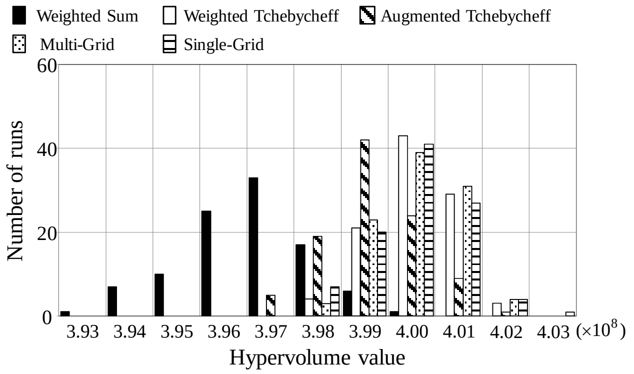


Figure. 8. Experimental results on the 2-500 problem.

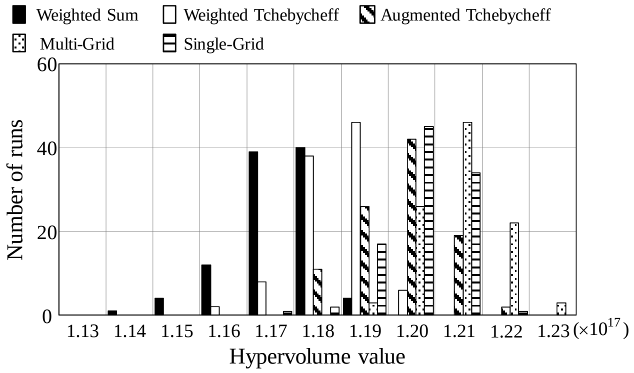


Figure. 9. Experimental results on the 4-500 problem.

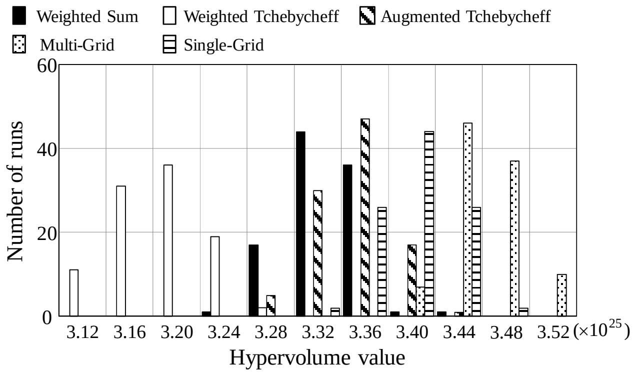


Figure. 10. Experimental results on the 6-500 problem.

In Fig. 10, the best results on the 6-500 problem were obtained from the multi-grid scheme. Since the population size in this variant of MOEA/D was twice as large as that of the other variants (due to the use of two complete grids of weight vectors), we further examined the performance of each variant of MOEA/D on the 6-500 problem using larger grids of weight vectors. For fair comparison of experimental results with different settings of the population size, we always used the total number of examined solutions (i.e., 200,000 solution evaluations) as the stopping condition throughout all computational experiments in this paper as shown in Section 3. Experimental results are summarized in Fig. 11. Independent of the size of grids of weight vectors, the best results were obtained from the multi-grid scheme in Fig. 11. We can also observe that much better results were obtained by the simultaneous use of the two scalarizing functions than their individual use in Fig. 11. The proposed idea also outperformed the augmented weighted Tchebycheff. It should be noted that the two implementation schemes of the proposed idea introduce no additional parameter whereas ρ was used as an additional parameter in the augmented weighted Tchebycheff.

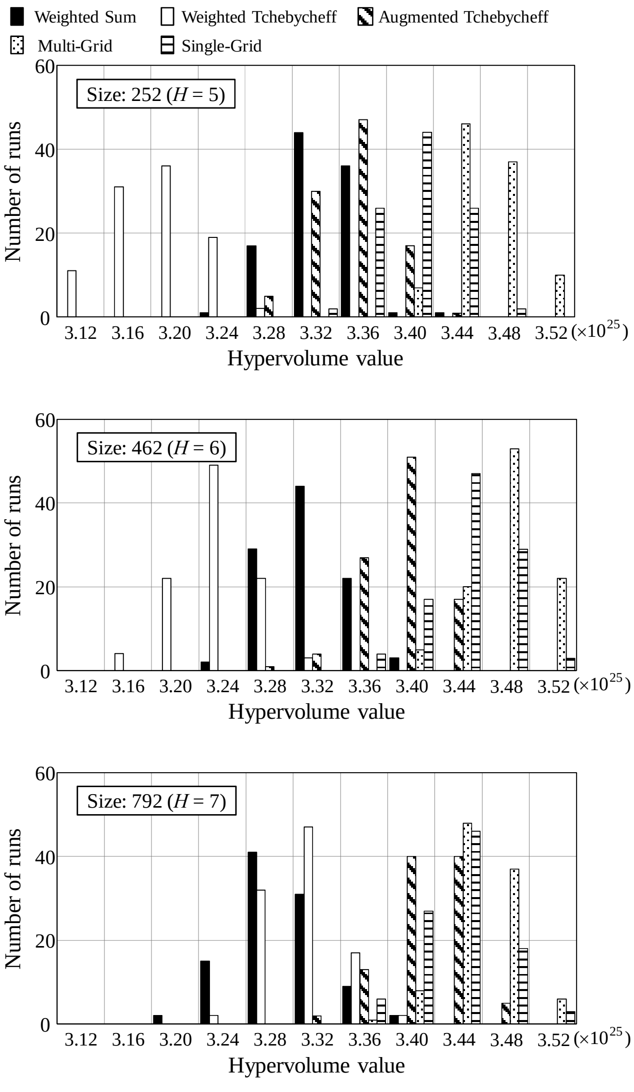


Figure. 11. Experimental results on the 6-500 problem for three grids of different size: 252 ($H = 5$), 462 ($H = 6$) and 792 ($H = 7$).

6. CONCLUDING REMARKS

In this paper, we proposed an idea of simultaneously using different types of scalarizing functions in MOEA/D. We also examined two implementation schemes of the proposed idea. One is to use multiple grids of weight vectors where each grid is used by a single scalarizing function. The other is to alternately assign a different scalarizing function to each weight vector in a single grid. The effectiveness of these implementation schemes was examined through computational experiments on multiobjective 0/1 knapsack problems with two, four and six objectives. Experimental results showed that the simultaneous use of the weighted sum and the weighted Tchebycheff outperformed their individual use in MOEA/D. Especially in the case of the six-objective 0/1 knapsack problem, much better results were obtained from our approach than the use of a single scalarizing function. These observations suggest the existence of the synergy effect by the use of different scalarizing functions.

One interesting observation, which is somewhat counter-intuitive, is that the increase in the population size did not always lead to the performance improvement of MOEA/D on the six-objective 0/1 knapsack problem except for the case with the weighted Tchebycheff function (see Fig. 11). It may be true that EMO algorithms need a large population when they are applied to many-objective problems. However, this statement does not mean that the performance of EMO algorithms can be always improved by increasing the population size when they are executed under the limited computation load.

Whereas we only examined the use of the weighted sum and the weighted Tchebycheff, our idea is applicable to the case with more than two scalarizing functions. It would be interesting to examine the performance of MOEA/D with more than two types of scalarizing functions. It would be also interesting to examine the use of multiple augmented weighted Tchebycheff functions with different parameter values of ρ. These interesting issues are left for future research topics.

7. REFERENCES

[1] Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. *Proc. of Foundations of Genetic Algorithm X* (2009) 87-102

[2] Beume, N., Naujoks, B., and Emmerich M. SMS-EMOA: multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 180, 3 (2007) 1653-1669

[3] Chang, P. C., Chen, S. H., Zhang, Q., and Lin, J. L. MOEA/D for flowshop scheduling problems. *Proc. of 2008 IEEE Congress on Evolutionary Computation* (2008) 1433-1438.

[4] Coello, C. A. C., and Lamont, G. B. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore (2004).

[5] Coello, C. A. C., van Veldhuizen, D. A., and Lamont, G. B. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer, Boston (2002).

[6] Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*. John Wiley & Sons, Chichester (2001).

[7] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation* 6, 2 (2002) 182-197.

[8] Figueira, J. R., Liefooghe, A., Talbi, E.-G., and Wierzbicki, A.P. A parallel multiple reference point approach for multi-

objective optimization. *European Journal of Operational Research* 205, 2 (2010) 390-400

[9] Friedrich, T., Horoba, C., and Neumann, F. Multiplicative approximations and the hypervolume indicator. *Proc. of 2009 Genetic and Evolutionary Computation Conference* (2009) 571-578.

[10] Goldberg, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading (1989).

[11] Hughes, E. J. Evolutionary many-objective optimization: Many once or one many?. *Proc. of 2005 IEEE Congress on Evolutionary Computation* (2005) 222-227.

[12] Hughes, E. J. MSOPS-II: A general-purpose many-objective optimizer. *Proc. of 2007 IEEE Congress on Evolutionary Computation* (2007) 3944-3951.

[13] Ikeda, K., Kita, H., and Kobayashi, S. Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?. *Proc. of 2001 IEEE Congress on Evolutionary Computation* (2001) 957-962.

[14] Ishibuchi, H., Doi, T., and Nojima, Y. Incorporation of scalarizing fitness functions into evolutionary multiobjective optimization algorithms. *Lecture Notes in Computer Science 4193: Parallel Problem Solving from Nature - PPSN IX*, Springer, Berlin (2006) 493-502.

[15] Ishibuchi, H., and Nojima, Y. Optimization of scalarizing functions through evolutionary multiobjective optimization. *Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007*, Springer, Berlin (2007) 51-65.

[16] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multi-objective evolutionary algorithm. *Proc of 5th International Conference on Evolutionary Multi-Criterion Optimization* (2009) 438-452.

[17] Ishibuchi, H., Sakane, Y., Tsukamoto, N., and Nojima, Y. Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations. *Proc. of 2009 IEEE International Conference on Systems, Man, and Cybernetics* (2009) 1820-1825.

[18] Ishibuchi, H., Tsukamoto, N., Hitotsuyanagi, Y., and Nojima, Y. Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems. *Proc. of 2008 Genetic and Evolutionary Computation Conference* (2008) 649-656.

[19] Jaszkiewicz, A. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem - A comparative experiment. *IEEE Trans. on Evolutionary Computation* 6, 4 (2002) 402-412.

[20] Jaszkiewicz, A. On the computational efficiency of multiple objective metaheuristics. The knapsack problem case study. *European Journal of Operational Research* 158, 2 (2004) 418-433.

[21] Khara, V., Yao, X., and Deb, K. Performance scaling of multi-objective evolutionary algorithms. *Lecture Notes in Computer Science 2632: Evolutionary Multi-Criterion Optimization - EMO 2003*, Springer, Berlin (2004) 367-390.

[22] Li, H., and Zhang, Q. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Trans. on Evolutionary Computation* 13, 2 (2009) 284-302.

[23] Murata, T., Ishibuchi, and H., Gen, M. Specification of genetic search directions in cellular multi-objective genetic algorithm. *Lecture Notes in Computer Science 1993: Evolutionary Multi-Criterion Optimization - EMO 2001*, Springer, Berlin (2001) 82-95.

[24] Peng, W., Zhang, Q., and Li, H. Comparison between MOEA/D and NSGA-II on the multi-objective travelling salesman problem. in Goh, C. K., Ong, Y. S., and Tan, K. C. (eds) *Multi-Objective Memetic Algorithms*. Springer, Berlin (2009) 309-324.

[25] Purshouse, R. C. and Fleming, P. J. On the evolutionary optimization of many conflicting objectives. *IEEE Trans. on Evolutionary Computation* 11, 6 (2007) 770-784.

[26] Tan, K. C., Khor, E. F., and Lee, T. H. *Multiobjective Evolutionary Algorithms and Applications*, Springer, Berlin (2005).

[27] Wagner, T., Beume, N., and Naujoks, B. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. *Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007*, Springer, Berlin (2007) 742-756.

[28] Zhang, Q., and Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. on Evolutionary Computation* 11, 6 (2007) 712-731.

[29] Zitzler, E., Brockhoff, D., and Thiele, L. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. *Lecture Notes in Computer Science 4403: Evolutionary Multi-Criterion Optimization - EMO 2007*, Springer, Berlin (2007) 862-876.

[30] Zitzler, E., and Künzli, S. Indicator-based selection in multiobjective search. *Lecture Notes in Computer Science 3242: Parallel Problem Solving from Nature - PPSN VIII*, Springer, Berlin (2004) 832-842.

[31] Zitzler, E., and Thiele, L. Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Trans. on Evolutionary Computation* 3, 4 (1999) 257-271.

[32] Zou, X., Chen, Y., Liu, M., and Kang, L. A new evolutionary algorithm for solving many-objective optimization problems. *IEEE Trans. on Systems, Man, and Cybernetics: Part B - Cybernetics* 38, 5 (2008) 1402-1412.