Many-Objective Optimization by Space Partitioning and Adaptive ϵ-Ranking on MNK-Landscapes

Hernán Aguirre[1,2] and Kiyoshi Tanaka[2]

[1] Fiber-Nanotech Young Researcher Empowerment Program
[2] Shinshu University, Faculty of Engineering,
4-17-1 Wakasato, Nagano, 380-8553 Japan
{ahernan,ktanaka}@shinshu-u.ac.jp

Abstract. This work proposes a method to search effectively on *many*-objective problems by instantaneously partitioning the objective space into subspaces and performing one generation of the evolutionary search in each subspace. The proposed method uses a partition strategy to define a schedule of subspace sampling, so that different regions of objective space could be emphasized at different generations. In addition, it uses an adaptive ϵ-ranking procedure to re-rank solutions in each subspace, giving selective advantage to some of the solutions initially ranked highest in the whole objective space. Adaptation works to keep the actual number of highest ranked solutions in each subspace close to a desired number. The performance of the proposed method is verified on MNK-Landscapes. Experimental results show that convergence and diversity of the solutions found can improve remarkably on $4 \leq M \leq 10$ objectives.

1 Introduction

Multiobjective evolutionary algorithms (MOEAs) [1,2] optimize simultaneously two or more objective functions, aiming to find a set of trade-off solutions in a single run of the algorithm. Most state of the art MOEAs use Pareto dominance within the selection procedure of the algorithm to rank solutions. Selection based on Pareto dominance is thought to be effective for problems with convex and non-convex fronts and has been successfully applied in two and three objectives problems.

Recently, there is a growing interest on applying MOEAs to solve *many*-objective optimization problems, i.e. problems with four or more objectives. However, current research reveals that the number of Pareto non-dominated solutions gets substantially larger as we increase the number of objectives of the problem [3,4]. Hence, ranking by Pareto dominance becomes coarser and too many solutions are assigned the same rank. This affects the effectiveness of selection, severely deteriorating the performance of MOEAs [5,6,7].

In this work, we propose a method to search on many-objective problems by instantaneously partitioning the objective space into subspaces and performing one generation of the evolutionary search in each subspace. Partitioning of the

M. Ehrgott et al. (Eds.): EMO 2009, LNCS 5467, pp. 407–422, 2009.

objective space into subspaces aims to instantaneously emphasize the search within smaller regions of objective space. The proposed method uses a partition strategy to define a schedule of subspace sampling, so that different regions could be emphasized at different generations. In addition, it uses an adaptive ϵ-ranking procedure to re-rank solutions in each subspace, giving selective advantage to some of the solutions initially ranked highest in the whole objective space, so that selection can put more emphasizes in exploitation. Adaptation in the re-ranking procedure works to keep the actual number of highest ranked solutions in each subspace close to a desired number. The combination of space partitioning, partitioning strategy, and adaptive ϵ-ranking allows to perform an effective search aiming to improve convergence and diversity of solutions on many-objective problems.

In this paper, we implement the proposed method using NSGA-II's framework [8]. We test the proposed method on MNK-Landscapes [3,4] with $4 \leq M \leq 10$ objectives, $N = 100$ bits, and $0 \leq K \leq 50$ epistatic interactions. Experimental results show that convergence and diversity of the solutions found can improve remarkably on $4 \leq M \leq 10$ objectives for all K.

2 Multiobjective Optimization Concepts and Definitions

Let us consider, without loss of generality, a maximization multiobjective problem with M objectives:

$$maximize \quad \boldsymbol{f}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})) \tag{1}$$

where $\boldsymbol{x} \in \mathcal{X}$ is a solution vector in the solution space $\mathcal{X}$, and $f_1, f_2, \cdots, f_M$ the M objective functions to be optimized.

Definition 1 (Objective space ϕ). *The objective space of the problem is determined by the set $\phi = \{f_1, f_2, \cdots, f_M\}$ of the M objective functions to be optimized.*

One dimensional comparison and Pareto optimality are two popular methods used to decide what solution to choose from a set of solutions. Yu [9] showed that these two methods are extreme cases in the entire domain of domination structures and that there are infinity valid methods lying between them, which suitability depends on how much information is known on the decision maker's preferences. Within the EMO community, these other domination structures are also known as relaxed forms of Pareto dominance and one method to implement them is ϵ-dominance [10]. Pareto dominance and ϵ-dominance concepts are of special relevance to this work and are defined as follows.

Definition 2 (Pareto dominance). *A solution $\boldsymbol{x}$ is said to Pareto dominate other solution $\boldsymbol{y}$ in the objective space ϕ if the two following conditions are satisfied:*

$$\begin{array}{ll} \forall f_m \in \phi & f_m(\boldsymbol{x}) \geq f_m(\boldsymbol{y}) \ \wedge \\ \exists f_m \in \phi & f_m(\boldsymbol{x}) > f_m(\boldsymbol{y}). \end{array} \tag{2}$$

Here, $\boldsymbol{x}$ dominates $\boldsymbol{y}$ is denoted by $\boldsymbol{f}(\boldsymbol{x}) \succeq \boldsymbol{f}(\boldsymbol{y})$.

Definition 3 (ϵ-dominance). *A solution $\boldsymbol{x}$ is said to ϵ-dominate other solution $\boldsymbol{y}$ in the objective space ϕ if the two following conditions are satisfied:*

$$\begin{array}{ll} \forall f_m \in \phi & (1+\epsilon)f_m(\boldsymbol{x}) \geq f_m(\boldsymbol{y}) \;\wedge \\ \exists f_m \in \phi & (1+\epsilon)f_m(\boldsymbol{x}) > f_m(\boldsymbol{y}). \end{array} \tag{3}$$

where $\epsilon > 0.0$. Here, $\boldsymbol{x}$ ϵ-dominates $\boldsymbol{y}$ is denoted by $\boldsymbol{f}(\boldsymbol{x}) \succeq^{\epsilon} \boldsymbol{f}(\boldsymbol{y})$.

Other important concepts we use to describe our algorithm are defined as follows.

Definition 4 (Subspace ψ). *A subspace ψ of ϕ is a lower dimension space that includes some of the functions in ϕ, i.e. $\psi \subset \phi$.*

Definition 5 (Non-overlapping subspaces). *Two subspaces $\psi_1 \subset \phi$ and $\psi_2 \subset \phi$ are said to be non-overlapping if they have no common objectives, i.e. $\psi_1 \cap \psi_2 = \emptyset$.*

Definition 6 (Space partition Ψ_{N_S}). *An space ϕ is said to be partitioned into N_S subspaces, denoted as Ψ_{N_S}, if all subspaces are non-overlapping and no objective function in ϕ is left unassigned to a subspace, i.e. $\Psi_{N_S} = \{\psi_1, \psi_2, \cdots, \psi_{N_S} \mid \psi_1 \cap \psi_2 \cdots \cap \psi_{N_S} = \emptyset \wedge \psi_1 \cup \psi_2 \cdots \cup \psi_{N_S} = \phi\}$.*

Definition 7 (Subspace ϵ-dominance). *A solution $\boldsymbol{x}$ is said to ϵ-dominate other solution $\boldsymbol{y}$ in the subspace ψ if:*

$$\begin{array}{ll} \forall f_m \in \psi & (1+\epsilon)f_m(\boldsymbol{x}) \geq f_m(\boldsymbol{y}) \;\wedge \\ \exists f_m \in \psi & (1+\epsilon)f_m(\boldsymbol{x}) > f_m(\boldsymbol{y}). \end{array} \tag{4}$$

Here, $\boldsymbol{x}$ ϵ-dominates $\boldsymbol{y}$ in the subspace ψ is denoted by $\boldsymbol{f}(\boldsymbol{x}) \succeq^{\epsilon}_{\psi} \boldsymbol{f}(\boldsymbol{y})$.

3 Method

3.1 Concept

In this section, we describe the proposed method to search on many-objective problems by space partitioning and adaptive ϵ-ranking. In the following, we call this method ϵ Ranking Multiobjective Optimizer (ϵR-EMO). The goal of ϵR-EMO is to find a set of solutions with good properties of convergence and diversity. To achieve its goal, ϵR-EMO first ranks solutions by Pareto dominance calculated in the whole objective space. Then, it instantaneously partitions the objective space into subspaces, re-ranks solutions for each subspace using an adaptive subspace-ϵ-ranking procedure, and performs one generation of the evolutionary search within each subspace. During the next cycle of the algorithm, parents and offspring from all subspaces will be joined together so that they will be ranked again in the whole objective space.

By partitioning the objective space into subspaces, we aim to instantaneously emphasize the search within smaller regions of objective space. At each generation, we don't search in all possible subspaces. Instead, we define a schedule of subspace sampling by using a partition strategy. Re-ranking of solutions by the adaptive subspace-ϵ-ranking aims to give selective advantage to some of the usually too many solutions assigned highest rank in a many-objective subspace, so that selection can put more emphasizes in exploitation. Adaptation in the re-ranking procedure works to keep the actual number of highest ranked solutions in each subspace close to a desired number. The combination of space partitioning, partitioning strategy, and adaptive subspace-ϵ-ranking, aims to effectively search on many-objective problems.

In the following we first explain the general flow of the proposed method using NSGA-II's framework [8] and then explain in detail its distinctive features.

Procedure 1. ϵR-EMO

```
Input: $N_S$, number of subspaces at each generation. $\alpha$, desired number of solutions
    with highest rank in each subspace (as a fraction of the entire parent population)
Output: $\mathcal{F}_1$, set of Pareto non-dominated solutions
 1: $\mathcal{P} \leftarrow \emptyset$, $\mathcal{Q} \leftarrow random$  // initialize parent $\mathcal{P}$ and offspring $\mathcal{Q}$ populations
 2: $\epsilon_1, \epsilon_2, \cdots, \epsilon_{N_S} \leftarrow 0.0$
 3: repeat
 4:    evaluation($\mathcal{Q}, \phi$)  // $\phi = \{f_1, f_2, \cdots, f_M\}$
 5:    $\mathcal{F} \leftarrow$ non-domination-sorting($\mathcal{P} \cup \mathcal{Q}$)  // $\mathcal{F} = \{\mathcal{F}_i\}$  $(i = 1, 2 \cdots, N_F)$
 6:    crowding-distance($\mathcal{F}$)
 7:    $\Psi_{N_S} \leftarrow$ subspace-partition($\phi$, $N_S$)  // $\Psi_{N_S} = \{\psi_1, \psi_2, \cdots, \psi_{N_S}\}$
 8:    $P \leftarrow \emptyset$, $Q \leftarrow \emptyset$
 9:    for $s = 1$ to $N_S$ do
10:        $\mathcal{F}^{\epsilon_s} \leftarrow$ subspace-$\epsilon$-ranking ($\psi_s$, $\epsilon_s$, $\mathcal{F}$)  // $\mathcal{F}^{\epsilon_s} = \{\mathcal{F}_j^{\epsilon_s}\}$  $(j = 1, 2 \cdots, N_F^{\epsilon_s})$
11:        $\epsilon_s \leftarrow$ adaptation($\epsilon_s$, $\alpha$, $|\mathcal{F}_1^{\epsilon_s}|$)  // adapt $\epsilon_s$ for the next generation
12:        $\mathcal{P}_s \leftarrow$ truncation($\mathcal{F}^{\epsilon_s}$)  // $|\mathcal{P}_s| = |\mathcal{P}|/N_S$, $|\mathcal{F}^{\epsilon_s}| = |\mathcal{P}| + |\mathcal{Q}|$
13:        $\mathcal{Q}_s \leftarrow$ recombination and mutation($\mathcal{P}_s$)  // $|\mathcal{Q}_s| = |\mathcal{Q}|/N_S$
14:        $\mathcal{P} \leftarrow \mathcal{P} \cup \mathcal{P}_s$, $\mathcal{Q} \leftarrow \mathcal{Q} \cup \mathcal{Q}_s$
15:    end for
16: until termination criterion is met
17: return $\mathcal{F}_1$
```

3.2 ϵR-EMO

ϵR-EMO implemented in NSGA-II's framework [8] is illustrated in Procedure 1. See that solutions are evaluated in all M objectives $\phi = \{f_1, f_2, \cdots, f_M\}$, ranked based on Pareto dominance, and assigned a crowding measure using non-domination sorting and crowding distance procedures [8], respectively (lines 4-6). After this initial ranking, solutions are classified in sets of non-dominated solutions $\mathcal{F} = \{\mathcal{F}_i\}$ $(i = 1, 2, \cdots, N_F)$. Next, the objective space ϕ is partitioned into N_S non-overlapping subspaces $\Psi_{N_S} = \{\psi_1, \psi_2, \cdots, \psi_{N_S}\}$ (line 7). Then, for each subspace ψ_s, solutions $\mathcal{F}$ are re-ranked and re-classified in $\mathcal{F}^{\epsilon_s} = \{\mathcal{F}_j^{\epsilon_s}\}$ $(i = 1, 2, \cdots, N_F^{\epsilon_s})$ using a subspace-ϵ-ranking procedure, where $N_F^{\epsilon_s} \geq N_F$, creating a finer grained ranking of solutions (line 10). Subspace-ϵ-ranking uses parameter ϵ_s to control the number of highest ranked individuals $|\mathcal{F}_1^{\epsilon_s}|$. Parameter ϵ_s is adapted every generation (line 11) to keep $|\mathcal{F}_1^{\epsilon_s}|$ close to a desired number. The parent population in each subspace $\mathcal{P}_s$ is obtained by truncating $\mathcal{F}^{\epsilon_s}$ based on rank and crowding distance (line 12). That is, groups of solutions $\mathcal{F}_j^{\epsilon_s}$ are assigned iteratively to $\mathcal{P}_s$ by rank order, starting with $\mathcal{F}_1^{\epsilon_s}$. If $\mathcal{F}_j^{\epsilon_s}$ overfills $\mathcal{P}_s$, crowding distance calculated in the whole space ϕ is used to choose the required number o solutions. Mating for recombination is carried out by binary tournament, where winners are decided by rank in the subspace ψ_s breaking ties by crowding distance in ϕ.

3.3 Subspace Partitioning

In our approach, we partition the M dimensional space $\phi = \{f_1, f_2, \cdots, f_M\}$ into N_S non-overlapping subspaces $\Psi_{N_S} = \{\psi_1, \psi_2, \cdots, \psi_{N_S}\}$. All subspaces have the same dimension $M_S = M/N_S$ in case $r = (M \mod N_S)$ is zero. Otherwise, r of the N_S subspaces have dimension $M_S = \lfloor M/N_S \rfloor + 1$ and the rest $M_S = \lfloor M/N_S \rfloor$. The number of all possible ways to partition ϕ into subspaces of dimension M_S is very large. In our approach, we don't explicitly search in all possible subspaces at each generation. Instead, we set N_S to a small value and define a schedule of subspace sampling by using a partitioning strategy. We investigate three strategies to partition ϕ. Namely, *random*, *shift*, and *fixed* partition strategies.

Random strategy randomly assigns objectives $f_i \in \phi$ to subspaces $\psi_s \in \Psi_{N_S}$. With this strategy, any possible M_S dimensional subspace of ϕ could be formed. However, it does not seek to correlate the s-th subspace ψ_s from generation t to the next.

Shift strategy, at the first generation, assigns deterministically objectives $f_i \in \phi$ to subspaces $\psi_s \in \Psi_{N_S}$, so that objectives assigned to a given ψ_s are ordered by objective index i. Then, in subsequent generations, the objective with highest index in the s-th subspace is shifted to the $((s+1) \mod N_S)$-th subspace, $\forall \psi_s \in \Psi_{N_S}$. This strategy correlates the s-th subspace from generation t to the next. In fact, subspace ψ_s at generation t overlaps with ψ_s at generation $t-1$ in all but one objective. However, not all possible M_S dimensional subspaces of ϕ could be formed.

Fixed strategy assigns deterministically objectives $f_i \in \phi$ to subspaces $\psi_s \in \Psi_{N_S}$ and keep the same assignment throughout the generations. With this strategy only N_S subspaces of ϕ could be formed.

These strategies would allow us to verify the impact of subspace sampling on the quality of solutions and the effect of subspace correlation from one generation to the next on the adaptation of ϵ for subspace-ϵ-ranking.

3.4 Adaptation of ϵ

In our method, solutions are re-ranked in each subspace by using a subspace-ϵ-ranking procedure in which the number of solutions assigned highest rank depends on the value set to ϵ (≥ 0) and on the instantaneous distribution of solutions in objective space (see below). Although it is difficult to tell in advance exactly how many solutions will be assigned highest rank for a given value of ϵ, we know that larger values of ϵ decrease the number of highest ranked solutions and vice versa. The algorithm takes advantage of this correlation to adapt ϵ at each generation in order to keep the actual number of highest ranked solutions close to a desired number [12]. The desired number of highest ranked solution in each subspace is specified by $\alpha \times |\mathcal{P}|$, where α is a parameter in the range [0.0,1.0] set by the user and $|\mathcal{P}|$ is the size of the entire parent population.

In our method, instead of using one ϵ for all subspaces, we adapt one ϵ_s for each one of the N_S subspaces ψ_s. Note that the actual combination of objectives that define ψ_s change with time, depending on the partition strategy. So, adaptation of ϵ_s reacts to the characteristics of the different instantaneous subspaces (actual combinations of f_i) assigned to ψ_s. Since the dimension of the subspace is strongly correlated to the value of ϵ that renders the desired number of highest ranked solutions $\alpha \times |P|$, when the space ϕ is partitioned we make sure that the dimension of the subspace ψ_s remains the same throughout the generations.

3.5 Subspace-ϵ-Ranking

Subspace-ϵ-ranking fine grains ranking of solutions initially ranked by Pareto dominance in the objective space ϕ, using a randomized ϵ-sampling procedure in the subspace $\psi \subset \phi$ that favors a good distribution of solutions based on dominance regions wider than conventional Pareto dominance. Subspace-ϵ-ranking extends ϵ-ranking [13], where ϵ-sampling acts on ϕ instead of ψ. In the following, we first explain ϵ-sampling and then subspace-ϵ-ranking.

ϵ-sampling assumes that there is a set of equally ranked solutions from which a subset should be chosen to give them selective advantage in order to proceed further with the evolutionary search. That is, ϵ-sampling acts as a decision making procedure, not to find a final solution, but to help selection of the evolutionary algorithm. Hence, the sampling heuristic must reflect criteria that favor an effective search. Here, the sample of solutions to be given selective advantage are obtained with the following criteria,

- Extreme solutions are always part of the sample.
- Each (not extreme) sampled solution is the sole sampled representative of its area of influence. The area of influence of the sampled solutions is determined by a domination region wider than Pareto dominance, i.e. ϵ-dominance.
- Sampling of (not extreme) solutions follows a random schedule.

The first criterion tries to push the search towards the optimum values of each fitness function, aiming to find non-dominated solutions in a wide area of objective space. The second criterion assures that only one solution in a given zone of objective space is given higher rank, trying to distribute the search effort more or less uniformly among the different zones represented in the actual population. The third criterion dynamically establishes the zones that are represented in the sample. Also, in the case that there are several solutions within each zone, it increases the likelihood that the sampled solutions that will be given higher rank are different from one generation to the next, increasing the possibility of exploring wider areas of objective and variable space.

Procedure 2 illustrates ϵ-sampling algorithm. Let us denote $\mathcal{A}$ the set of solutions that have been assigned the same rank based on conventional Pareto dominance, for example by applying non-domination sorting [8]. ϵ-sampling returns the sampled solutions $\mathcal{S} \subset \mathcal{A}$ that will be given selective advantage as well as the set of solutions $\mathcal{D}^\epsilon$ to be demoted. See that extreme solutions are the first to be assigned to the sample $\mathcal{S}$ (lines 1,2). Then, one by one, solutions are randomly chosen and included in $\mathcal{S}$ (lines 4-6), whereas solutions that lie in the wider domination region of the randomly picked solution are assigned to $\mathcal{D}^\epsilon$ (lines 7,8). Note that subspace ϵ-dominance $\boldsymbol{f}(z) \succeq_\psi^\epsilon \boldsymbol{f}(y)$ is used. Fig. 1 **(a)** illustrates the application of ϵ-sampling on the set of solutions $\mathcal{A} = \mathcal{F}_1$. The numbers close to the solutions represents the random schedule in which solutions are sampled (0 means extreme solutions, which are all selected at once).

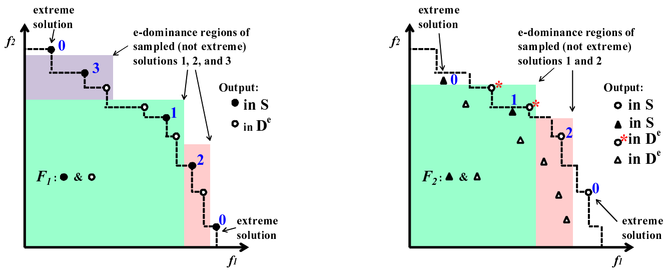
Procedure 2. ϵ-sampling (ψ, ϵ, $\mathcal{A}$, $\mathcal{S}$, $\mathcal{D}^\epsilon$)

Input: Subspace ψ, ϵ-dominance factor ϵ and a set of solutions $\mathcal{A}$
Output: $\mathcal{S}$ and $\mathcal{D}^\epsilon$ ($\mathcal{S} \cup \mathcal{D}^\epsilon = \mathcal{A}$). $\mathcal{S}$ contains the sample of solutions from $\mathcal{A}$, whereas $\mathcal{D}^\epsilon$ contains ϵ-dominated solutions in ψ

```
1: $\mathcal{X} \leftarrow \{x \in \mathcal{A} \mid f_m(x) = \max(f_m(\cdot)), \forall f_m \in \psi\}$
2: $\mathcal{S} \leftarrow \mathcal{X}$ , $\mathcal{A} \leftarrow \mathcal{A} \setminus \mathcal{X}$, $\mathcal{D}^\epsilon \leftarrow \emptyset$
3: while $\mathcal{A} \neq \emptyset$ do
4:     $r \leftarrow rand()$   // $1 \leq r \leq |\mathcal{A}|$
5:     $z \leftarrow r$-th solution $\in \mathcal{A}$
6:     $\mathcal{S} \leftarrow \mathcal{S} \cup \{z\}$
7:     $\mathcal{Y} \leftarrow \{y \in \mathcal{A} \mid \boldsymbol{f}(z) \succeq_\psi^\epsilon \boldsymbol{f}(y), z \neq y\}$
8:     $\mathcal{D}^\epsilon \leftarrow \mathcal{D}^\epsilon \cup \mathcal{Y}$
9:     $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\{z\} \cup \mathcal{Y}\}$
10: end while
11: return
```

The ϵ-sampling procedure works on a set of equally ranked solutions, however within a population there could be several sets of such solutions (each set with a different rank). Here, we explain subspace-ϵ-ranking to re-rank all possible sets of equally ranked solutions using ϵ-sampling.

Subspace-ϵ-ranking is applied at each generation for each subspace after non-domination sorting to reclassify the sets $\mathcal{F}_i$ ($i = 1, \cdots, N_F$). Procedure 3 describes subspace-ϵ-ranking algorithm. See that the reclassified sets $\mathcal{F}_j^{\epsilon_s}$ ($j = 1, \cdots, N_F^{\epsilon_s}$) now contains only the sample of solutions $\mathcal{S} \subset \mathcal{F}_i$ found by ϵ-sampling (lines 8,9). Also, see that solutions $\mathcal{D}^\epsilon$, which are not part of the sample (line 8) are demoted by joining them with solutions of a lower ranked set in the next iteration of the loop (line 4). Thus, $\mathcal{F}_1^{\epsilon_s}$ contains some of the solutions initially ranked first, but $\mathcal{F}_j^{\epsilon_s}$, $j > 1$, can contain solutions that initially were assigned to sets with different ranks. This gives chance to lateral diversity present



(a) $\mathcal{F}_1$ (b) $\mathcal{F}_2 \cup \mathcal{D}^{\epsilon}_{\mathcal{F}_1}$

Fig. 1. ϵ-sampling on (a) the set of solutions initially ranked first and (b) on the set of solutions initially ranked second joined with solutions demoted from the first set

in the initial ranking of solutions and can punish highly crowded solutions even if they are initially ranked first by conventional Pareto dominance.

Procedure 3. subspace-ϵ-ranking (ψ_s, ϵ_s, $\mathcal{F}$, $\mathcal{F}^{\epsilon_s}$)

Input: Subspace ψ_s, ϵ-dominance factor ϵ_s and solutions $\mathcal{F}$ classified in fronts $\mathcal{F}_i$ ($i = 1, \cdots, N_F$) by non-domination sorting

Output: $\mathcal{F}^{\epsilon_s}$, solutions re-classified in groups $\mathcal{F}_j^{\epsilon_s}$ ($j = 1, \cdots, N_F^{\epsilon_s}$)

```
$\mathcal{D}^\epsilon \leftarrow \emptyset$, $i \leftarrow 1$, $j \leftarrow 1$
repeat
    if $i \leq N_F$ then
        $\mathcal{A} \leftarrow \mathcal{F}_i \cup \mathcal{D}^\epsilon$ , $i \leftarrow i + 1$
    else
        $\mathcal{A} \leftarrow \mathcal{D}^\epsilon$
    end if
    $\epsilon$-sampling($\psi$, $\epsilon_s$, $\mathcal{A}$, $\mathcal{S}$, $\mathcal{D}^\epsilon$)
    $\mathcal{F}_j^{\epsilon_s} \leftarrow \mathcal{S}$, $j \leftarrow j + 1$
until $\mathcal{D}^\epsilon = \emptyset$
return
```

Fig. 1 illustrates ϵ-ranking calling on ϵ-sampling to re-rank the set $\mathcal{F}_1$ of solutions initially ranked first and the set $\mathcal{F}_2$ of solutions ranked second joined with the demoted solutions $\mathcal{D}^\epsilon$ from $\mathcal{F}_1$. The example illustrates the application of ϵ-sampling to a 2 dimensional objective space ϕ. When ϵ-sampling is applied to a subspace $\psi \subset \phi$, the non-dominated solutions in ϕ projected in the subspace ψ (assuming a 2 dimensional subspace) would look similar to Fig. 1 **(b)**. ϵ-sampling will be applied to all projected solutions.

4 Test Problems, Performance Measures and Parameters

4.1 Multiobjective MNK-Landscapes

In this work we test the performance of the algorithms on multiobjective MNK-Landscapes. A multiobjective MNK-Landscape [3,4] is defined as a vector function mapping binary strings into real numbers $\boldsymbol{f}(\cdot) = (f_1(\cdot), f_2(\cdot), \cdots, f_M(\cdot))$:

$\mathcal{B}^N \rightarrow \Re^M$, where M is the number of objectives, $f_i(\cdot)$ is the i-th objective function, $\mathcal{B} = \{0, 1\}$, and N is the bit string length. $\boldsymbol{K} = \{K_1, \cdots, K_M\}$ is a set of integers where K_i $(i = 1, 2, \cdots, M)$ is the number of bits in the string that epistatically interact with each bit in the i-th landscape. Each $f_i(\cdot)$ can be expressed as an average of N functions as follows

$$f_i(\boldsymbol{x}) = \frac{1}{N} \sum_{j=1}^{N} f_{i,j}(x_j, z_1^{(i,j)}, z_2^{(i,j)}, \cdots, z_{K_i}^{(i,j)}) \tag{5}$$

where $f_{i,j} : \mathcal{B}^{K_i+1} \rightarrow \Re$ gives the fitness contribution of bit x_j to $f_i(\cdot)$, and $z_1^{(i,j)}, z_2^{(i,j)}, \cdots, z_{K_i}^{(i,j)}$ are the K_i bits interacting with bit x_j in the string $\boldsymbol{x}$. The fitness contribution $f_{i,j}$ of bit x_j is a number between [0.0, 1.0] drawn from a uniform distribution. Thus, each $f_i(\cdot)$ is a non-linear function of $\boldsymbol{x}$ expressed by a Kauffman's NK-Landscape model of epistatic interactions [11]. In addition, it is also possible to arrange the epistatic pattern between bit x_j and the K_i other interacting bits. That is, the distribution $D_i = \{random, nearest\ neighbor\}$ of K_i bits among N. Thus, M, N, $\boldsymbol{K} = \{K_1, K_2, \cdots, K_M\}$, and $\boldsymbol{D} = \{D_1, D_2, \cdots, D_M\}$, completely specify a multiobjective MNK-Landscape.

4.2 Performance Measures

In this work, we use the hypervolume $\mathcal{H}$ and coverage $\mathcal{C}$ measures [14] to evaluate and compare the performance of the algorithms. The measure $\mathcal{H}$ calculates the volume of the M-dimensional region in objective space enclosed by a set of non-dominated solutions and a dominated reference point. Let $\mathcal{A}$ be a set of non-dominated solutions. The hypervolume of $\mathcal{A}$ can be expressed as

$$\mathcal{H}(\mathcal{A}) = \cup_{i=1}^{|\mathcal{A}|}(\mathcal{V}_i - \cap_{j=1}^{i-1}\mathcal{V}_i\mathcal{V}_j) \tag{6}$$

where $\mathcal{V}_i$ is the hypervolume rendered by the point $\boldsymbol{x}_i \in \mathcal{A}$ and the reference point. In this work, the reference point is set to $[0.0, \cdots, 0.0]$. Given two sets of non-dominated solutions $\mathcal{A}$ and $\mathcal{B}$, if $\mathcal{H}(\mathcal{A}) > \mathcal{H}(\mathcal{B})$ then set $\mathcal{A}$ can be considered better on convergence and/or diversity of solutions. To calculate $\mathcal{H}$, we use Fonseca et al. [15] algorithm, which significantly reduces computational time.

The coverage $\mathcal{C}$ measure [14] provides complementary information on convergence. Let us denote $\mathcal{A}$ and $\mathcal{B}$ the sets of non-dominated solutions found by two algorithms. $\mathcal{C}(\mathcal{A}, \mathcal{B})$ gives the fraction of solutions in $\mathcal{B}$ that are dominated at least by one solution in $\mathcal{A}$. More formally,

$$\mathcal{C}(\mathcal{A}, \mathcal{B}) = \frac{|\ \{\boldsymbol{b} \in \mathcal{B} | \exists \boldsymbol{a} \in \mathcal{A} : \boldsymbol{f}(\boldsymbol{a}) \succeq \boldsymbol{f}(\boldsymbol{b})\}\ |}{|\ \mathcal{B}\ |}. \tag{7}$$

$\mathcal{C}(\mathcal{A}, \mathcal{B}) = 1.0$ indicates that all solutions in $\mathcal{B}$ are dominated by solutions in $\mathcal{A}$, whereas $\mathcal{C}(\mathcal{A}, \mathcal{B}) = 0.0$ indicates that no solution in $\mathcal{B}$ is dominated by solutions in $\mathcal{A}$. Since usually $\mathcal{C}(\mathcal{A}, \mathcal{B}) + \mathcal{C}(\mathcal{B}, \mathcal{A}) \neq 1.0$, both $\mathcal{C}(\mathcal{A}, \mathcal{B})$ and $\mathcal{C}(\mathcal{B}, \mathcal{A})$ are required to understand the degree to which solutions of one set dominate solutions of the other set.

4.3 Parameters

In this work, we test the performance of the algorithm on MNK-Landscapes with $4 \leq M \leq 10$ objectives, $N = 100$ bits, number of epistatic interactions $K = \{0, 1, 3, 5, 10, 15, 25, 35, 50\}$ $(K_1, \cdots, K_M = K)$, and *random* epistatic patterns among bits for all objectives $(D_1, \cdots, D_M = random)$. Results presented below show the average performance of the algorithms on 50 different problems randomly generated for each combination of M, N and K. In the plots, error bars show 95% confidence intervals on the mean.

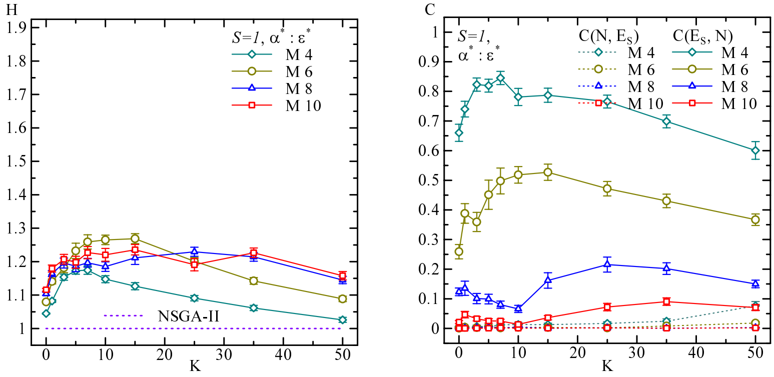
In the following sections we analyze results by ϵR-EMO, comparing them with results by conventional NSGA-II. The algorithms use parent and offspring populations of size $|\mathcal{P}| = |\mathcal{Q}| = 100$, two point crossover for recombination with rate $p_c = 0.6$, and bit flipping mutation with rate $p_m = 1/N$ per bit. The number of evaluations is set to 3×10^5. We study the performance of ϵR-EMO setting the number of subspaces to $N_S = \{1, 2\}$, varying the parameter α. For $N_S = 1$ (no subspace partitioning, $\Psi_{N_S} = \{\psi_1 = \phi\}$), we set $\alpha = \{1.0, 0.7, 0.5, 0.3\}$ so that the desired number of solutions with highest rank after subspace-ϵ-ranking is $\alpha \times |\mathcal{P}| = \{100, 70, 50, 30\}$, respectively. For $N_S = 2$ subspaces (subspace partition $\Psi_{N_S} = \{\psi_1, \psi_2\}$), we set $\alpha = \{0.5, 0.35, 0.25, 0.15\}$ so that $\alpha \times |\mathcal{P}| = \{50, 35, 25, 15\}$, respectively, in each of the two subspaces.

5 Experimental Results and Discussion

5.1 Performance by ϵR-EMO with No Objective Space Partitioning

In this section, we discuss the performance of ϵR-EMO when no objective space partitioning is considered $(N_S = 1)$, setting the fraction between desired number of highest ranked individuals and population size to $\alpha = \alpha^*$ that achieves maximum hypervolume $\mathcal{H}$. Fig. 2 (a) shows the average ratio $\frac{\mathcal{H}(E)}{\mathcal{H}(N)}$, where E and N denote the set of solutions found by ϵR-EMO and conventional NSGA-II, respectively. Thus, a ratio greater than 1.0 indicates better $\mathcal{H}$ by ϵR-EMO than conventional NSGA-II. As a reference, we include a horizontal line to represent the $\mathcal{H}(N)$ values normalized to 1.0. From this figure, we can see that ϵR-EMO can significantly improve $\mathcal{H}$ on $4 \leq M \leq 10$ objectives problems, for all values of K (up to 27% improvement). Note that improvements on $\mathcal{H}$ become larger as we increase the number of objectives M from 4 to 6, whereas improvements on $\mathcal{H}$ are similarly high for $8 \leq M \leq 10$. Due to space limitations, we include results for $M = \{4, 6, 8, 10\}$ only and not for $M = \{5, 7, 9\}$.

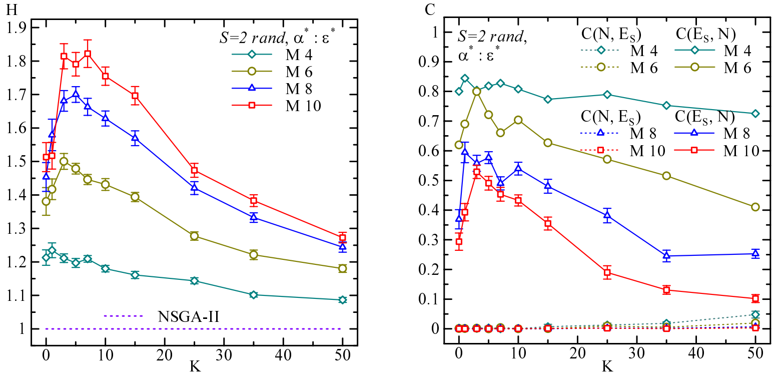
Improvements on $\mathcal{H}$ can be due to solutions with better convergence, better diversity, or both. To complement the analyzes of results on $\mathcal{H}$ we also present results using the $\mathcal{C}$ measure. Fig. 2 (b) shows the average $\mathcal{C}$ values between conventional NSGA-II and ϵR-EMO set with α^*. From this figure, we can be see that $\mathcal{C}(N, E)$ is close to 0.0 for most K and M. This indicates that there are almost no solutions by conventional NSGA-II that dominate solutions by ϵR-EMO. On the other hand, the values of $\mathcal{C}(E, N)$ are very high for 4 objectives (in the range 0.60-0.85) and reduce gradually as we increase M up to 10 objectives (in the



a) Normalized $\mathcal{H}$

b) $\mathcal{C}$ measure

Fig. 2. Normalized $\mathcal{H}$ and $\mathcal{C}$ between NSGA-II and ϵR-EMO when no objective space division ($S = 1$) is considered. ϵR-EMO is set to α^* that achieves maximum $\mathcal{H}(E)$.



a) Normalized $\mathcal{H}$

b) $\mathcal{C}$ measure

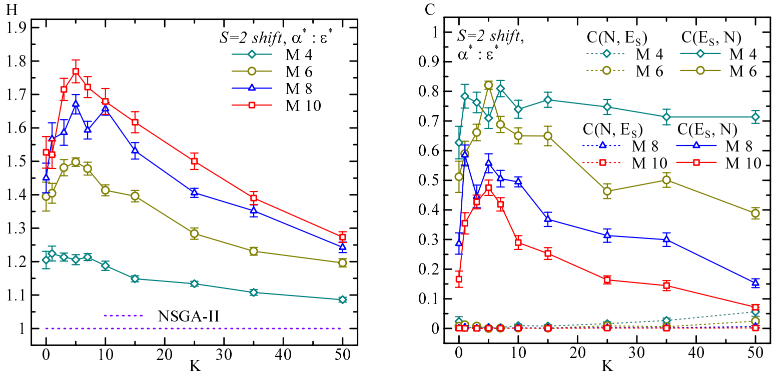
Fig. 3. Results by NSGA-II and ϵR-EMO partitioning the objective space in two subspaces ($S = 2$) using *random* strategy and setting α^* that achieves maximum $\mathcal{H}(E)$

range 0.01-0.08). This suggests that a better convergence of solutions contributes to the increases of $\mathcal{H}$ by ϵR-EMO on $M = 4$ problems. As we increase M, gains on diversity gradually become more significant than gains on convergence as the reason for the significant improvement of $\mathcal{H}$ on $6 \leq M \leq 10$.

5.2 Performance by ϵR-EMO with Objective Space Partitioning

In this section we analyze the performance of ϵR-EMO partitioning instantaneously the objective space into two subspaces using the *random*, *shift*, and *fixed* partition strategies introduced in section 3.

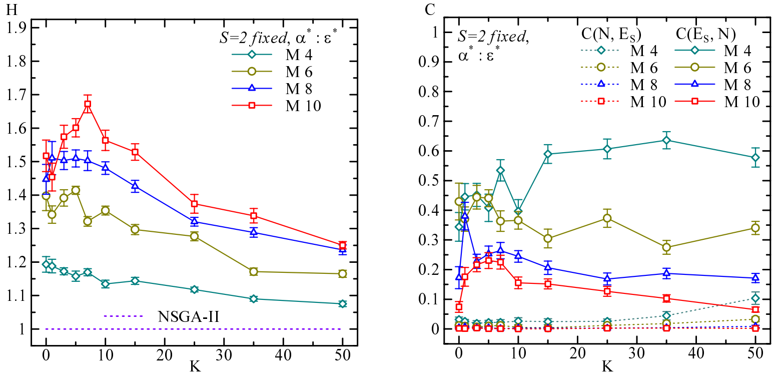
First, we show results by the *random* partitioning strategy in Fig. 3. Looking at Fig. 3 (a) and comparing with Fig. 2 (a), we can see that ranking on subspaces using a *random* strategy leads to a remarkable improvement on $\mathcal{H}$ for all values



a) Normalized $\mathcal{H}$

b) $\mathcal{C}$ measure

Fig. 4. Results by NSGA-II and ϵR-EMO partitioning the objective space in two subspaces ($S = 2$) using *shift* strategy and setting α^* that achieves maximum $\mathcal{H}(E)$



a) Normalized $\mathcal{H}$

b) $\mathcal{C}$ measure

Fig. 5. Results by NSGA-II and ϵR-EMO partitioning the objective space in two subspaces ($S = 2$) using *fixed* strategy and setting α^* that achieves maximum $\mathcal{H}(E)$

of M and K (up to 82.5% improvement). Note that the increase on $\mathcal{H}$ gets bigger with the number of objectives M. Looking at at Fig. 3 (b) and comparing with Fig. 2 (b), we can see that $\mathcal{C}(E, N)$ also increases for any value of K and M, whereas $\mathcal{C}(N, E)$ remains close to zero. That is, convergence also improves substantially.

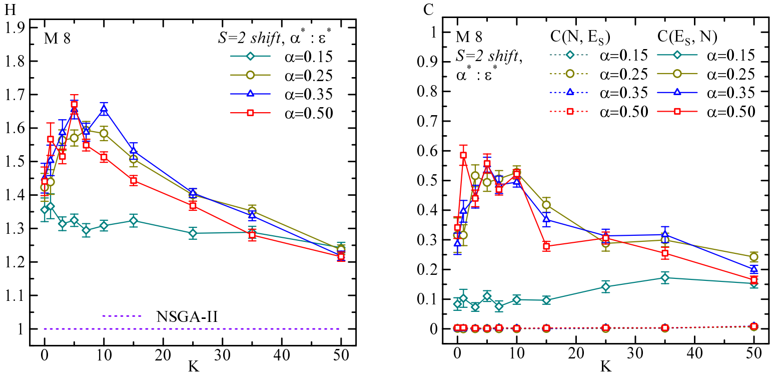
Next, we discuss results by the *shift* partition strategy shown in Fig. 4 (a) and (b). From these figures note that the *shift* strategy also leads to a remarkable improvement on $\mathcal{H}$ and $\mathcal{C}$. Comparing the *shift* and *random* strategies, the latter leads to slightly better results than the former especially for $M \geq 8$ and $K \leq 15$. As mentioned above, the *random* strategy can sample any possible subspace of ϕ, whereas the *shift* strategy can sample most but not all subspaces of ϕ. The number of subspaces unable to sample the *fixed* strategy increase with the dimension of the objective space, especially if we keep constant the number of subspaces. Better results by the *random* strategy suggests that sampling all possible subspaces becomes relevant as the number of objectives increase.

Results by the *fixed* partition strategy are shown in Fig. 5 (a) and (b). See that the *fixed* strategy leads to smaller $\mathcal{H}$ and $\mathcal{C}(E, N)$ than the *random* and *shift* partition strategies. Comparing to ϵR-EMO with no subspace partitioning, the *fixed* strategy leads to higher $\mathcal{H}$ on all M but with smaller $\mathcal{C}(E, N)$ on $M \leq 6$. The *fixed* strategy only explores N_S of all possible subspaces of ϕ and it is seems not an appropriate strategy to achieve best performance on both convergence and diversity of solutions.

Finally, note that as we increase K (non-linearity of the problem) improvements on both $\mathcal{H}$ and $\mathcal{C}$ reduce regardless of the partition strategy. This suggests that in addition to better ranking strategies, we should also look into ways to improve recombination and mutation to achieve better performance on highly non-linear problems.

5.3 Analysis of α

In this section we analyze the parameter α that determines the desired number of highest ranked solution in each subspace. As an example, Fig. 6 shows $\mathcal{H}$ and $\mathcal{C}$ results achieved by different settings of α on $M = 8$ objectives landscapes partitioning the objective space in $N_S = 2$ subspaces using *shift* strategy. From this figure, see that $\alpha \geq 0.25$ (at least 50%of the parent population in each subspace is given highest rank) leads to high performance, whereas results by $\alpha = 0.15$ are clearly lower on $M = 8$ objectives landscapes. Analyzing performance by $\alpha \geq 0.25$, see that setting α to 0.35 or 0.25 leads to best performance for most K, both on $\mathcal{H}$ and $\mathcal{C}$. However, see that setting α to 0.5 could give highest performance especially on small K. Although there is not an absolute winner among $\alpha \geq 0.25$ values, it is important to note that subspace partitioning's



a) Normalized $\mathcal{H}$ b) $\mathcal{C}$ measure

Fig. 6. Results by NSGA-II and ϵR-EMO partitioning the objective space in two subspaces ($S = 2$) using *shift* strategy and varying α on $M = 8$ objectives

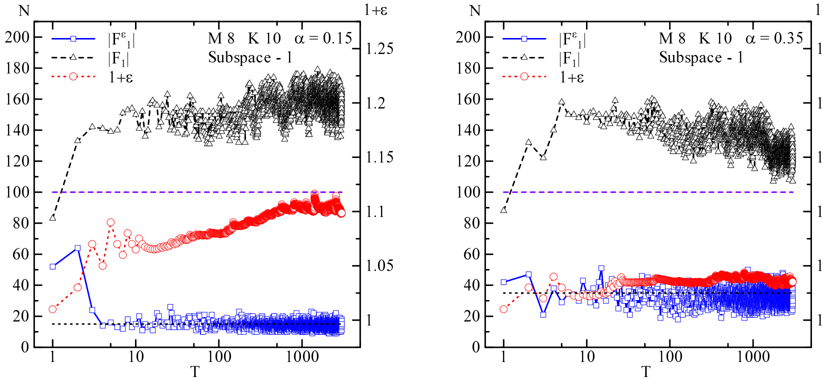


Fig. 7. Adaptation of ϵ in ϵR-EMO for $\alpha = 0.15$ and $\alpha = 0.35$ on $M = 8$ objectives and $K = 10$ epistatic bits

lower bound performance ($\min \mathcal{H} \wedge \min \mathcal{C}(E, N), \forall \alpha \in \{0.5, 0.35, 0.25\}$) is by far better than the performance by no subspace partitioning (see Fig. 6 and compare with Fig. 2). Analyzing our data for other values of M, in general, we see that performance by $\alpha = 0.25$ is better than 0.35 when the number of objectives decrease to $M = 6$ and $M = 4$; whereas performance by 0.5 and 0.35 is better than 0.25 when we increase M to 10 objectives. As a rule of thumb, when the space is partitioned into 2 subspaces, $\alpha = 0.25$ works well on $M = 4$ and $M = 6$, $\alpha = 0.35$ on $M = 8$, and $\alpha = 0.5$ on $M = 10$.

Fig. 7 illustrates adaptation of ϵ_s for $\alpha = 0.15$ and $\alpha = 0.35$ for one of the two subspaces (the adaptation trend in the other subspace is similar) in a $M = 8$ and $K = 10$ landscape. The horizontal dashed line at $N = 100$ indicates the size of the overall parent population $|P|$ and the horizontal dotted line the desired number of individuals $\alpha \times |\mathcal{P}|$ with highest rank in the subspace. From these figures note that the number of non-dominated individuals $|\mathcal{F}_1|$ (considering all objectives) exceeds $\mathcal{P}$ since the initial generations. See also that the adaptive mechanism appropriately varies ϵ_s throughout the generations so that after subspace-ϵ-ranking the number of individuals $|\mathcal{F}_1^{\epsilon_s}|$ with highest rank in the subspace is kept around the desired number $\alpha \times |\mathcal{P}|$.

6 Conclusions

In this work, we have proposed a method to search on many-objective problems by instantaneously partitioning the objective space into subspaces and performing one generation of the evolutionary search in each subspace. The proposed method uses a partition strategy to define the schedule of subspace sampling and an adaptive re-ranking method that uses a randomized sampling procedure to increase selection probabilities of some of the too many solutions assigned highest rank in a many-objective subspace. We tested the performance of the proposed method on MNK-Landscapes with $4 \leq M \leq 10$ objectives, $N = 100$

bits and $0 \leq K \leq 50$ epistatic interactions, showing that both convergence and diversity of the obtained solutions can improve remarkably on problems with $4 \leq M \leq 10$ objectives for any level of epistatic interactions K. We also showed that uniformly sampling all possible subspaces throughout the generations leads to better performance.

As future works, we would like to study the effects of larger population sizes and more than two subspaces. Also, we should compare the proposed method with other approaches for many-objective optimization.

Acknowledgment

This study was performed through Special Coordination Funds for Promoting Science and Technology of the Ministry of Education, Culture, Sports, Science and Technology, of the Japanese Government.

References

1. Deb, K.: Multi-Objective Optimization using Evolutionary Algorithms. John Wiley & Sons, Chichester (2001)
2. Coello, C., Van Veldhuizen, D., Lamont, G.: Evolutionary Algorithms for Solving Multi-Objective Problems. Kluwer Academic Publishers, Boston (2002)
3. Aguirre, H., Tanaka, K.: Insights on Properties of Multiobjective MNK-Landscapes. In: Proc. 2004 IEEE Congress on Evolutionary Computation, pp. 196–203. IEEE Service Center, Los Alamitos (2004)
4. Aguirre, H., Tanaka, K.: Working Principles, Behavior, and Performance of MOEAs on MNK-Landscapes. European Journal of Operational Research 181(3), 1670–1690 (2007)
5. Purshouse, R., Fleming, P.: Conflict, harmony, and independence: Relationships in evolutionary multi-criterion optimisation. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 16–30. Springer, Heidelberg (2003)
6. Aguirre, H., Tanaka, K.: Selection, drift, recombination, and mutation in multiobjective evolutionary algorithms on scalable MNK-landscapes. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 355–369. Springer, Heidelberg (2005)
7. Hughes, E.J.: Evolutionary Many-Objective Optimization: Many Once or One Many? In: Proc. 2005 IEEE Congress on Evolutionary Computation, vol. 1, pp. 222–227. IEEE Service Center, Los Alamitos (2005)
8. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, KanGAL report 200001 (2000)
9. Yu, P.L.: Cone Convexity, Cone Extreme Points, and Nondominated Solutions in Decision Problems with Multiobjectives. Journal of Optimization Theory and Applications 14(3), 319–377 (1974)
10. Laumanns, M., Thiele, L., Deb, K., Zitzler, E.: Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. Evolutionary Computation 10(3), 263–282 (2002)

11. Kauffman, S.A.: The Origins of Order: Self-Organization and Selection in Evolution. Oxford University Press, Oxford (1993)
12. Aguirre, H., Tanaka, K.: Adaptive ϵ-Ranking on MNK-Landscapes. In: Proc. 2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision Making (2009)
13. Aguirre, H., Tanaka, K.: Robust Optimization by ϵ-Ranking On High Dimensional Objective Spaces. In: Li, X., Kirley, M., Zhang, M., Green, D., Ciesielski, V., Abass, H.A., Michalewicz, Z., Hendtlass, T., Deb, K., Tan, K.C., Branke, J., Shi, Y. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 421–431. Springer, Heidelberg (2008)
14. Zitzler, E.: Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, PhD thesis, Swiss Federal Institute of Technology, Zurich (1999)
15. Fonseca, C., Paquete, L., López-Ibáñez, M.: An Improved Dimension-sweep Algorithm for the Hypervolume Indicator. In: Proc. 2006 IEEE Congress on Evolutionary Computation, pp. 1157–1163. IEEE Service Center, Los Alamitos (2006)