Indicator-Based Differential Evolution Using Exclusive Hypervolume Approximation and Parallelizatio for Multi-core Processors

Kiyoharu Tagawa
School of Science and
Engineering, Kinki University
Osaka 577-8501, Japan
tagawa@info.kindai.ac.jp

Hidehito Shimizu
Panasonic Electronic
Devices Co., Ltd
Osaka 571-8506, Japan
simizu@jp.panasonic.com

Hiroyuki Nakamura
Panasonic Electronic
Devices Co., Ltd
Osaka 571-8506, Japan
nakamura@jp.panasonic.com

ABSTRACT

A new Multi-Objective Evolutionary Algorithm (MOEA) based on Differential Evolution (DE), i.e., Indicator-Based DE (IBDE) is proposed. IBDE employs a strategy of DE for generating a series of offspring. In order to evaluate the quality of each individual in the population, IBDE uses the exclusive hypervolume as an indicator function. A fast algorithm called Incremental Hypervolume by Slicing Objectives (IHSO) has been reported for calculating the exclusive hypervolume. However, the computational time spent by IHSO increases exponentially with the number of objectives and considered individuals. Therefore, an exclusive hypervolume approximation, in which IHSO can be also used effectively, is proposed. Furthermore, it is proven that the proposed exclusive hypervolume approximation gives an upper bound of the accurate exclusive hypervolume. The procedure of IHSO is parallelized by using the multiple threads of the Java language. By using the parallelized IHSO, not only the exclusive hypervolume but also the exclusive hypervolume approximation can be calculated concurrently on a multi-core processor. By the results of numerical experiments and statistical tests conducted on test problems, the usefulness of the proposed approach is demonstrated.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Heuristic methods; G.1.6 [**Optimization**]: Global optimization; D.2 [**Software**]: Software Engineering

General Terms

Algorithms

Keywords

Differential evolution, multi-objective optimization, hypervolume approximation, parallelization

GECCO'11, July 12–16, 2011, Dublin, Ireland.
Copyright 2011 ACM 978-1-4503-0557-0/11/07 ...$10.00.

1. INTRODUCTION

Differential Evolution (DE) is a relatively recent Evolutionary Algorithm (EA) proposed by R. Storn and K. Price [22], which was designed to solve single-objective optimization problems over continuous decision spaces. DE is a simple yet powerful algorithm that outperforms typical EAs such as Genetic Algorithm (GA), Evolutionary Strategy (ES) and Particle Swarm Optimization (PSO) on many single-objective optimization problems [22, 21]. Therefore, DE has been used for numerous real-world applications [21].

For trying to translate its good properties exhibited in the single-objective optimization problem to the multi-objective case, DE has been adapted to solve multi-objective optimization problems in several ways. In the early approaches [21], only the concept of Pareto dominance was used for the survival selection in which an offspring was selected only if it weakly dominated its parent. Many subsequent approaches such as NSDE [15], GDE2 [18], GDE3 [19] and NSDE-DCS [16] used the non-dominated sorting and/or the crowding distance metric in their survival selections. Recently, the survival selections used by three state-of-the-art Multi-Objective EAs (MOEAs) were introduced individually into DE [26]. Then, from the results of numerical experiments and statistical tests, it was shown that the three DE-based MOEAs outperformed their counterparts, or the basic MOEAs, on the majority of examined test problems.

Pareto dominance-based MOEAs usually work very well on two-objective optimization problems. However, the search ability of those MOEAs decreases as the number of objectives increases [14, 20]. Therefore, Pareto dominance-based MOEAs cannot be applied effectively to many-objective optimization problems that have more than four objectives. A promising approach for solving many-objective optimization problems is a class of Indicator-based EAs (IBEAs) that use an indicator function to measure the quality of each individual in the current population [29, 27]. The hypervolume is the only indictor known to be strictly monotonic with respect to Pareto dominance and thereby guaranteeing that the Pareto-optimal front achieves the maximum hypervolume possible, while any worse set will be assigned a worse indicator value. The hypervolume was originally proposed to quantitatively compare the outcomes of different MOEAs [30]. Several IBEAs that employ the hypervolume as the indicator function have been proposed [10, 7], but their main drawback is the extreme overhead for the hypervolume calculation. Even though dexterous algorithms for calculating the hypervolume have been reported [28, 12], their computational times increase exponentially with the number of objectives. Furthermore, it is expected that no polynomial algorithm exists for the problem of computing the hypervolume [5]. Therefore, several hypervolume approximations have been proposed. There are polynomial estimation algorithms for approximating the hypervolume based on Monte Carlo sampling [6, 1]. Besides, an achievement function with uniformly distributed weight vectors is proposed for approximating the hypervolume [17]. However, the accuracy of their estimations has a trade-off relationship with the computational time determined by their control parameters, namely the number of sampling points and weight vectors.

In this paper, a new DE-based MOEA named Indicator-Based DE (IBDE) is proposed. IBDE employs a strategy of DE for generating a series of offspring. After that, comparing each offspring with its parent based on Pareto dominance, IBDE selects either or both of the two individuals for the member of a tentative population. For selecting the member of the next population from the tentative population, IBDE uses the non-dominated sorting as a ranking criterion. Furthermore, for selecting some members from a part of the tentative population classified into the same rank, the exclusive hypervolume that is defined by a difference between two hypervolumes is used as an indicator function. A fast algorithm called Incremental Hypervolume by Slicing Objectives (IHSO) has been reported for calculating the exclusive hypervolume directly [3]. However, the computational time spent by IHSO also increases exponentially with the number of objectives and considered individuals. Therefore, an exclusive hypervolume approximation is proposed. Furthermore, it is proven that the exclusive hypervolume approximation gives an upper bound of the accurate exclusive hypervolume. By using the exclusive hypervolume approximation as the indicator function of IBDE, an iterative selection scheme becomes realistically applicable for choosing the member of the next population from a part of the tentative population classified into the same rank.

Recently, multi-core processors, which have more than one Central Processing Unit (CPU), have been introduced widely into personal computers. Therefore, in order to utilize the additional CPUs to execute costly application programs, concurrent implementations of them have been paid attention to [4]. Because EAs including DE maintain a lot of individuals manipulated competitively in the population, EAs have a parallel and distributed nature intrinsically. Consequently, some concurrent implementations of EAs have been reported for multi-core processors [25, 23]. In the program of IBDE, the procedure of IHSO is parallelized based on the multiple threads provided by the Java language. By using the parallelized IHSO, not only the exclusive hypervolume but also the proposed exclusive hypervolume approximation can be calculated concurrently on a personal computer equipped with a multi-core processor.

The rest of this paper is organized as follows. Section 2 describes a classic DE and a basic strategy of DE. Section 3 proposes the exclusive hypervolume approximation and IBDE. Section 4 describes how the calculation of IHSO can be parallelized for multi-core processors. Section 5 reports some numerical experiments to demonstrate the proposed approach. The results of numerical experiments are also verified by using statistical tests. Section 6 concludes this paper and discusses some possibilities for future work.

2. DIFFERENTIAL EVOLUTION (DE)

2.1 Representation

As we have mentioned earlier, DE [22] is used to solve the single-objective optimization problem formulated as

$$\left[\begin{array}{ll} \text{minimize} & f(\boldsymbol{x}) = f(x_1, \cdots, x_D) \\ \text{sub. to} & \underline{x}_j \le x_j \le \overline{x}_j,\ j = 1, \cdots, D. \end{array}\right. \tag{1}$$

Candidate solutions of (1) are called individuals. Besides, DE holds N_P individuals in the population for each generation G. Therefore, as well as conventional real-coded GAs, the i-th individual $\boldsymbol{x}_i^G$ $(i = 1, \cdots, N_P)$ included in the population of the generation G is represented as

$$\boldsymbol{x}_i^G = (x_{1,i}^G, \cdots, x_{j,i}^G, \cdots, x_{D,i}^G) \tag{2}$$

where, $x_{j,i}^G \in \mathbb{R}$ and $\underline{x}_j \le x_{j,i}^G \le \overline{x}_j$ $(j = 1, \cdots, D)$.

2.2 Strategy of DE

Differential mutation is a unique genetic operator of DE. Furthermore, a series of three genetic operators, namely reproduction selection, differential mutation and crossover, is used to generate offspring. The series of three genetic operators is called the strategy [22]. Even though various strategies have been proposed for DE [21, 11], a basic strategy named "DE/rand/1/bin" is described and used in this paper. That is because the basic strategy is powerful enough for solving real-world applications [21] and employed by some DE-based MOEAs such as GDE3 [19].

For each individual $\boldsymbol{x}_i^G$ $(i = 1, \cdots, N_P)$ in the population, which is also called the target vector, three different individuals, say $\boldsymbol{x}_{r1}$, $\boldsymbol{x}_{r2}$ and $\boldsymbol{x}_{r3}$ $(i \ne r1 \ne r2 \ne r3)$, are selected randomly from the current population. Then a new individual $\boldsymbol{u}_i^G = (u_{1,i}^G, \cdots, u_{j,i}^G, \cdots, u_{D,i}^G)$, which is also called the trial vector, is generated from the above four individuals by using the differential mutation and the crossover. In case of "DE/rand/1/bin", the procedure is described as

$$\left[\begin{array}{l} \text{for } (j = 1;\ j \le D;\ j = j + 1)\ \{ \\ \quad \text{if } (\text{rand}[0,\ 1] < C_R\ \vee\ j = j_r)\ \{ \\ \qquad u_{j,i}^G = x_{j,r1}^G + S_F\,(x_{j,r2}^G - x_{j,r3}^G); \\ \quad \}\ \text{else}\ \{ \\ \qquad u_{j,i}^G = x_{j,i}^G; \\ \quad \} \\ \} \end{array}\right. \tag{3}$$

where, rand[0, 1] is the random number generator that returns a uniformly distributed random number from within the range between 0 and 1. $j_r \in [1,\ D]$ is a randomly chosen index, which ensures that the trial vector $\boldsymbol{u}_i^G$ differs from the target vector $\boldsymbol{x}_i^G$ at least one element. The scale factor $S_F \in (0,\ 1+]$ and the crossover rate $C_R \in [0,\ 1]$ are control parameters specified by the user in advance.

If an element $u_{j,i}^G$ comes out of the range $[\underline{x}_j,\ \overline{x}_j]$ as a result of the procedure in (3), it is returned as follows:

$$u_{j,i}^G = \left\{\begin{array}{l} x_{j,r1}^G + \text{rand}[0,\ 1]\,(\underline{x}_j - x_{j,r1}^G) \text{ iff } u_{j,i}^G < \underline{x}_j, \\ x_{j,r1}^G + \text{rand}[0,\ 1]\,(\overline{x}_j - x_{j,r1}^G) \text{ iff } u_{j,i}^G > \overline{x}_j. \end{array}\right. \tag{4}$$

2.3 Procedure of DE

The procedure of the classic DE [22] can be described as follows. Since the classic DE is based on the generational model, two populations, namely the old one $\boldsymbol{x}_i^G \in \mathbf{P}^G$ and the new one $\boldsymbol{x}_i^{G+1} \in \mathbf{P}^{G+1}$, are used. For the stopping condition, the maximum number of generations G_{max} is specified.

[Classic DE]

Step 1 Randomly generate N_P individuals $\boldsymbol{x}_i^G \in \mathbf{P}^G$ as an initial population. Set the generation $G = 0$.

Step 2 If $G = G_{max}$ holds, output the best individual of the current population $\mathbf{P}^G$ and terminate.

Step 3 For each individual $\boldsymbol{x}_i^G \in \mathbf{P}^G$ $(i = 1, \cdots, N_P)$, execute everything from Step 3.1 to Step 3.2.

Step 3.1 Generate the trial vector $\boldsymbol{u}_i^G$ by (3) and (4).

Step 3.2 If $f(\boldsymbol{u}_i^G) \leq f(\boldsymbol{x}_i^G)$ holds then add $\boldsymbol{u}_i^G$ to $\mathbf{P}^{G+1}$ as $\boldsymbol{x}_i^{G+1} = \boldsymbol{u}_i^G$, otherwise $\boldsymbol{x}_i^{G+1} = \boldsymbol{x}_i^G$.

Step 4 Replace the old population $\boldsymbol{x}_i^G \in \mathbf{P}^G$ by the new one $\boldsymbol{x}_i^{G+1} \in \mathbf{P}^{G+1}$ as $\boldsymbol{x}_i^G = \boldsymbol{x}_i^{G+1}$ $(i = 1, \cdots, N_P)$.

Step 5 Set generation $G = G + 1$ and return to Step 2.

3. PROPOSED APPROACH

3.1 Problem and Definitions

Multi-objective optimization problems can be stated as

$$\left[\begin{array}{ll} \text{minimize} & F(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \cdots, f_M(\boldsymbol{x})) \\ \text{sub. to} & \underline{x}_j \leq x_j \leq \overline{x}_j, \; j = 1, \cdots, D. \end{array}\right. \tag{5}$$

The decision space $\Omega \subseteq \mathbb{R}^D$ is defined as a set of the solutions $\boldsymbol{x} \in \mathbb{R}^D$ that satisfy the boundary condition in (5). $F : \Omega \to \mathbb{R}^M$ consists of M real-valued objective functions and $\mathbb{R}^M$ is called the objective space. Let $\boldsymbol{x}, \boldsymbol{z} \in \Omega$, $\boldsymbol{x}$ is said to dominate $\boldsymbol{z}$ if and only if $f_m(\boldsymbol{x}) \leq f_m(\boldsymbol{z})$ for every $m \in \{1, \cdots, M\}$ and $f_n(\boldsymbol{x}) < f_n(\boldsymbol{z})$ for at least one index $n \in \{1, \cdots, M\}$. On the other hand, $\boldsymbol{x}$ is said to weakly dominate $\boldsymbol{z}$ if and only if $f_m(\boldsymbol{x}) \leq f_m(\boldsymbol{z})$ for every $m \in \{1, \cdots, M\}$. A solution $\boldsymbol{x}^\star \in \Omega$ is Pareto optimal to (5) if there is no solution $\boldsymbol{x} \in \Omega$ that dominates $\boldsymbol{x}^\star$.

A set of all the Pareto optimal solutions $\boldsymbol{x}^\star \in \Theta \subseteq \Omega$ is called the Pareto-optimal set. Besides, the Pareto front $F(\Theta) \subseteq \mathbb{R}^M$ is defined as $F(\Theta) = \{F(\boldsymbol{x}) \mid \boldsymbol{x} \in \Theta\}$. The approximation set $\Phi \subseteq \Omega$ is a set of non-dominated solutions of a limited number. In other words, every $\boldsymbol{x} \in \Phi$ is not dominated by any other $\boldsymbol{z} \in \Phi$. The approximation front $F(\Phi) \subseteq \mathbb{R}^M$ is also defined as $F(\Phi) = \{F(\boldsymbol{x}) \mid \boldsymbol{x} \in \Phi\}$. The purpose of MOEAs is to find an approximation set Φ that approximates the Pareto-optimal set Θ fairly well.

Let $B(\boldsymbol{x}, \boldsymbol{r}) \subseteq \mathbb{R}^M$ be a rectangular region shaped by $F(\boldsymbol{x}) \in \mathbb{R}^M$ $(\boldsymbol{x} \in \Phi)$ and a reference point $\boldsymbol{r} \in \mathbb{R}^M$. Furthermore, a region $B(\Phi, \boldsymbol{r})$ is composed as follows:

$$B(\Phi, \boldsymbol{r}) = \bigcup_{\boldsymbol{x} \in \Phi} B(\boldsymbol{x}, \boldsymbol{r}) \tag{6}$$

Now, the hypervolume $H(\Phi, \boldsymbol{r})$, which is used to measure the quality of an approximation set Φ, is defined as

$$H(\Phi, \boldsymbol{r}) = vol(B(\Phi, \boldsymbol{r})) \tag{7}$$

where, $vol(B) \in \mathbb{R}$ denotes the volume of a region $B \subseteq \mathbb{R}^M$.

3.2 Indicator-Based DE

The procedure of the proposed IBDE is described as follows. Besides the old $\boldsymbol{x}_i^G \in \mathbf{P}^G$ and the new $\boldsymbol{x}_i^{G+1} \in \mathbf{P}^{G+1}$ populations, a tentative population $\boldsymbol{z}_i^G \in \mathbf{Q}^G$ is used.

[IBDE]

Step 1 Randomly generate N_P individuals $\boldsymbol{x}_i^G \in \mathbf{P}^G$ as an initial population. Set the generation $G = 0$.

Step 2 If $G = G_{max}$ holds, output the set of non-dominated individuals $\Phi \subseteq \mathbf{P}^G$ with 1st rank and terminate.

Step 3 For each individual $\boldsymbol{x}_i^G \in \mathbf{P}^G$ $(i = 1, \cdots, N_P)$, execute everything from Step 3.1 to Step 3.3.

Step 3.1 Generate the trial vector $\boldsymbol{u}_i^G$ by (3) and (4).

Step 3.2 If $\boldsymbol{u}_i^G$ weakly dominates $\boldsymbol{x}_i^G$ then add $\boldsymbol{u}_i^G$ to $\mathbf{Q}^G$ as $\boldsymbol{z}_i^G = \boldsymbol{u}_i^G$, otherwise $\boldsymbol{z}_i^G = \boldsymbol{x}_i^G$.

Step 3.3 If $\boldsymbol{z}_i^G \neq \boldsymbol{u}_i^G$ in Step 3.2 and $\boldsymbol{x}_i^G$ does not dominate $\boldsymbol{u}_i^G$ then add $\boldsymbol{u}_i^G$ to $\mathbf{Q}^G$ as $\boldsymbol{z}_q^G = \boldsymbol{u}_i^G$ $(q > N_P)$.

Step 4 By using the non-dominated sorting contrived originally for NSGA-II [8], give a rank to each $\boldsymbol{z}_i^G \in \mathbf{Q}^G$.

Step 5 Select $\boldsymbol{x}_i^{G+1} \in \mathbf{P}^{G+1}$ $(i = 1, \cdots, N_P)$ from $\boldsymbol{z}_i^G \in \mathbf{Q}^G$ in the increasing order of the rank.

Step 6 If some $\boldsymbol{x}_i^{G+1} \in \mathbf{P}^{G+1}$ have to be selected from $\Phi \subseteq \mathbf{Q}^G$, or from a set of non-dominated solutions with the same rank, evaluate an indicator function for each $\boldsymbol{z}_i^G \in \Phi$. Eliminate extra $\boldsymbol{z}_i^G$ from Φ based on the indicator function. Then add the rest of Φ to $\mathbf{P}^{G+1}$.

Step 7 Replace the old population $\boldsymbol{x}_i^G \in \mathbf{P}^G$ by the new one $\boldsymbol{x}_i^{G+1} \in \mathbf{P}^{G+1}$ as $\boldsymbol{x}_i^G = \boldsymbol{x}_i^{G+1}$ $(i = 1, \cdots, N_P)$.

Step 8 Set generation $G = G + 1$ and return to Step 2.

Because some individuals are discarded in Step 3.2, the size of the tentative population is limited to

$$N_P \leq |\mathbf{Q}^G| \leq 2\,N_P \tag{8}$$

In Step 6 of IBDE, for each objective function, the individuals $\boldsymbol{z}_i^G \in \Phi$ with the smallest function values $f_m(\boldsymbol{z}_i^G)$ in Φ are assigned an infinite indicator function value.

3.3 Selection Scheme of IBDE

In Step 6 of the above IBDE, the exclusive hypervolume [3] can be used as an indicator function. The exclusive hypervolume is defined for each solution $\boldsymbol{x} \in \Phi$ of (5) as

$$EH(\boldsymbol{x}, \Phi, \boldsymbol{r}) = H(\Phi, \boldsymbol{r}) - H(\Phi \setminus \boldsymbol{x}, \boldsymbol{r}) \tag{9}$$

The exclusive hypervolume $EH(\boldsymbol{x}, \Phi, \boldsymbol{r})$ denotes how much additional hypervolume we get by adding $\boldsymbol{x}$ to Φ.

There are the following two schemes for eliminating k extra individuals from Φ based on the indicator function.

One shot: Evaluate the indicator function for each $\boldsymbol{z}_i^G \in \Phi$. Then eliminate k worst individuals from Φ.

Iterative: Repeat the following procedures k times.

1. Evaluate the indicator function for each $\boldsymbol{z}_i^G \in \Phi$.
2. Eliminate the worst individual $\boldsymbol{z}_i^G$ from Φ.

The good result can be expected by using the iterative approach [2]. That is because the contribution of a solution $\boldsymbol{x} \in \Phi$ to $H(\Phi, \boldsymbol{r})$ may be changed if another one $\boldsymbol{z} \in \Phi$ is eliminated from Φ. However, even though a dexterous algorithm called IHSO [3] has been reported for calculating the exclusive hypervolume directly, the computational time spent by IHSO increases exponentially with M and $|\Phi|$. Therefore, the iterative selection is not realistic for IBDE if the exclusive hypervolume is used as the indicator function.

As an alternative indicator function used by IBDE, we propose the exclusive hypervolume approximation as

$$EH2(\boldsymbol{x}, \Phi, \boldsymbol{r}) = \min_{\boldsymbol{z}\in\Phi\setminus\boldsymbol{x}} \{ EH(\boldsymbol{x}, \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) \} \tag{10}$$

The exclusive hypervolume approximation in (10) also uses IHSO for calculating $EH(\boldsymbol{x}, \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r})$. Even though IHSO has to be executed $|\Phi|-1$ times, it is expected that the computational time is reduced drastically. That is because only two individuals are considered by IHSO at a time. Furthermore, the proposed exclusive hypervolume approximation in (10) gives an upper bound of the accurate exclusive hypervolume in (9) as shown in the following theorem.

THEOREM 1. *Let Φ be an approximation set of (5) and $\boldsymbol{x} \in \Phi$. Furthermore, $\boldsymbol{r} \in \mathbb{R}^M$ is a reference point, then*

$$EH2(\boldsymbol{x}, \Phi, \boldsymbol{r}) \geq EH(\boldsymbol{x}, \Phi, \boldsymbol{r})$$

PROOF. See appendix. □

Now we present the following three variants of the proposed IBDE that differ in their selection schemes:

IBDE: In Step 6 of IBDE, the one shot selection scheme is used with the exclusive hypervolume.

IBDE2: In Step 6 of IBDE, the one shot selection is used with the exclusive hypervolume approximation.

IBDE2R: In Step 6 of IBDE, the iterative selection is used with the exclusive hypervolume approximation.

4. PARALLELIZATION OF IHSO

Multi-core processors, which have more than one Central Processing Unit (CPU), have been introduced widely into personal computers. Therefore, in order to utilize the additional CPUs to calculate $EH(\boldsymbol{z}_i^G, \Phi, \boldsymbol{r})$ $(i = 1, \cdots, |\Phi|)$ iteratively, a concurrent program of IHSO is developed. The concurrent program of IHSO, which is called the parallelized IHSO, can be modified easily for $EH2(\boldsymbol{z}_i^G, \Phi, \boldsymbol{r})$.

A program is said to be concurrent if it can support two or more tasks in process at the same time [4]. The concurrent program performs multiple tasks in parallel if it is executed on a multi-core processor. Therefore, it can be expected that the execution time of an algorithm is reduced by using the concurrent program on the multi-core processor.

The parallelized IHSO is designed in accordance with the master and slave mode. Then the parallelized IHSO is coded by the Java language, which is a very popular language supporting multiple threads. The main program of the parallelized IHSO, which corresponds to the master, evokes N_T $(N_T \geq 1)$ threads at a time. Each thread corresponds to a slave and calculates $EH(\boldsymbol{z}_i^G, \Phi, \boldsymbol{r})$ for a specified $\boldsymbol{z}_i^G \in \Phi$. The set of individuals Φ has been stored in a common memory shared among all CPUs. Then N_T threads or less are executed in parallel by using several CPUs. However, we cannot control the distribution of threads to CPUs. Furthermore, the computational time of each $EH(\boldsymbol{z}_i^G, \Phi, \boldsymbol{r})$ is not uniform. Therefore, in the proposed parallelized IHSO, the tasks for calculating $EH(\boldsymbol{z}_i^G, \Phi, \boldsymbol{r})$ $(i = 1, \cdots, |\Phi|)$ are assigned dynamically to the threads. The pseudocode of the parallelized IHSO can be described as follows:

```
// Master part
for all $n$ in parallel do {
    for ($n = 0$; $n < N_T$; $n++$) {
        Thread($n$);
    }
}

// Slave part
Thread($n$) {
    $i$ = GetIndex();
    while ($i < |\Phi|$) {
        Calculate $EH(\boldsymbol{z}_i^G, \Phi, \boldsymbol{r})$ by IHSO;
        $i$ = GetIndex();
    }
}
```

Each Thread(n) is assigned to one thread and gets a series of the individuals' indexes dynamically. `GetIndex()` denotes an exclusive function that returns an unique index of $\boldsymbol{z}_i^G \in \Phi$ at a time in ascending order such as $i = 0, 1, 2, \cdots$.

5. EXPERIMENTS

5.1 Test Problems

In order to assess the performance of the proposed approach, four scalable test problems, namely DTLZ1, DTLZ2, DTLZ3 and DTLZ4 [9], are employed. By using the scalable test problems, we can define the multi-objective problem with any number of decision variables and objectives.

Furthermore, in order to measure the optimality of a solution $\boldsymbol{x} \in \Omega$ for the above test problems, Convergence Measure (CM) is defined. CM gives the distance between a solution $\boldsymbol{x} \in \Omega$ and the nearest $\boldsymbol{x}^\star \in \Theta$. Therefore, the smaller CM is, the higher the solution's quality is. For every Pareto optimal one $\boldsymbol{x}^\star \in \Theta$, $CM(\boldsymbol{x}^\star) = 0$ holds.

First of all, CM of DTLZ1 is given as follows:

$$CM(\boldsymbol{x}) = |\sum_{m=1}^{M} f_m(\boldsymbol{x}) - 0.5| \tag{11}$$

Next, CM of DTLZ2 $\sim$ 4 is given as follows:

$$CM(\boldsymbol{x}) = |\sum_{m=1}^{M} f_m(\boldsymbol{x})^2 - 1| \tag{12}$$

Every program is coded by the Java language and executed on a personal computer equipped with a multi-core processor (CPU: Intel® Core$^{\text{TM}}$i7 @3.33[GHz]; OS: Microsoft Windows XP). The multi-core processor has four cores that can respectively manipulate two threads at the same time.

5.2 Comparison of EH and EH2

The sequence of the individuals sorted by using the exclusive hypervolume approximation $EH2$ was compared to that sorted by using the exclusive hypervolume EH. First of all, 100 Pareto-optimal solutions $\boldsymbol{x}^\star \in \Theta$ of DTLZ1 were generated randomly. Then the 100 solutions were sorted respectively by using $EH2$ and EH. Figure 1 plots out the

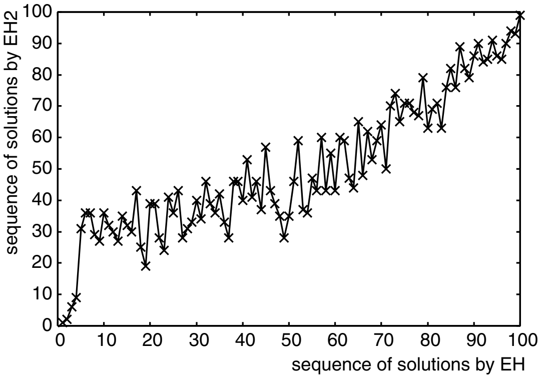

Figure 1: Solutions of DTLZ1 ($M = 5$)

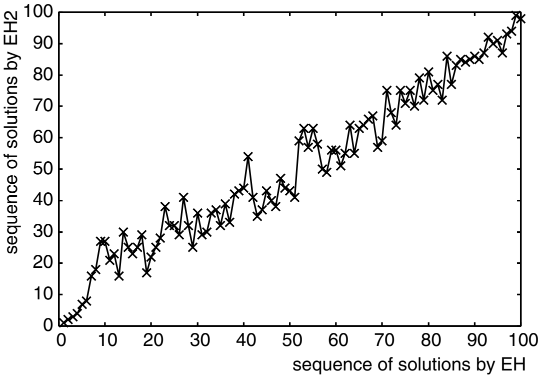

Figure 2: Solutions of DTLZ2 ($M = 5$)

two sequences of the 100 solutions $\boldsymbol{x}^\star \in \Theta$ of DTLZ1 with $M = 5$ objectives averaged 10 runs. Similarly, $EH2$ is compared with EH on DTLZ2. Figure 2 also plots out the two sequences of the 100 solutions $\boldsymbol{x}^\star \in \Theta$ of DTLZ2 with $M = 5$ averaged 10 runs. From the results of several numerical experiments conducted on DTLZ1 and DTLZ2, a clear linear relation can be observed between the two sequences of the solutions $\boldsymbol{x}^\star \in \Theta$ sorted by using $EH2$ and EH regardless of the kind of problems or the number of objectives.

By using a non-parametric statistical test called Spearman's rank correlation coefficient, the results of the above numerical experiments were verified. Table 1 summarizes the result of the non-parametric statistical test, namely Pearson's correlation coefficient (γ) between the two sequences of the solutions sorted respectively by $EH2$ and EH, t-static (t) and the risk (α) that is the probability to reject the null hypothesis when it is true. Symbol "▽" means that the risk α is less than 0.01. From the result in Table 1, the correlation between the two sequence was confirmed. Therefore, we can say that there is not a significant difference between two evaluations of solutions based on $EH2$ and EH.

Table 1: Statistical test of correlation

problem	DTLZ1		DTLZ2	
M	2	5	2	5
γ	0.988	0.916	0.989	0.976
t	65.17	22.72	68.14	44.94
α	▽	▽	▽	▽

Table 2: Hypervolume

problem	DTLZ1		DTLZ2	
M	2	5	2	5
EH	1.245	1.243	1.487	1.487
$EH2$	1.245	1.237	1.487	1.486

Table 3: Computational times of EH and EH2

(a) DTLZ1 ($M = 2$) [ms]

N_T	1	2	4	8
EH	16	16	12	10
$EH2$	40	28	19	18

(b) DTLZ1 ($M = 5$) [ms]

N_T	1	2	4	8
EH	1921	1121	806	699
$EH2$	128	78	62	53

Next, the hypervolume of a set of solutions $\Phi \subseteq \Theta$ selected by using $EH2$ was compared to that of Φ selected by using EH. First of all, 200 Pareto-optimal solutions $\boldsymbol{x}^\star \in \Theta$ of DTLZ1 were generated randomly. Then the top 100 solutions $\boldsymbol{x}^\star \in \Phi \subseteq \Theta$ were selected from Θ by using $EH2$ and EH. Similarly, the top 100 solutions were selected from 200 Pareto-optimal solutions $\boldsymbol{x}^\star \in \Phi \subseteq \Theta$ of DTLZ2 by using $EH2$ and EH. Table 2 shows the values of $H(\Phi, \boldsymbol{r})$ averaged 10 runs for the two test problems. From the result in Table 2, we can say that there is not so much difference between the hypervolumes brought by $EH2$ and EH.

Finally, the computational time for calculating $EH2$ was compared to that of EH. Table 3 shows the computational times for sorting the above 200 Pareto-optimal solutions $\boldsymbol{x}^\star \in \Theta$ of DTLZ1 based on $EH2$ and EH. Both $EH2$ and EH are calculated by using the parallelized IHSO with N_T threads. In case of $M = 2$ objectives, the computational times of EH are little shorter than those of $EH2$. Besides, the effect of the parallelization cannot be confirmed so much. On the other hand, in case of $M = 5$ objectives, the computational times of $EH2$ are much shorter than those of EH. Furthermore, the computational times of EH and $EH2$ are reduced obviously as the number of threads is increased.

5.3 Comparison of MOEAs

Besides the proposed three variants of IBDE, namely IBDE, IBDE2 and IBDE2R, GDE3 [19] was also applied to the above four test problems. GDE3 is a remarkable DE-based MOEA. The performance of DE has been evaluated through various test problems [19, 20]. Incidentally, GDE3 has been applied to a real-world application [24]. GDE3 is similar to the proposed IBDE except that GDE uses the crowding distance instead of the indicator function. For each of the four test problems, the objectives and decision variables were specified respectively as $(M, D) = (2, 7)$ and $(M, D) = (5, 10)$. The same control parameters were used for every MOEA: $S_F = 0.5$, $C_R = 0.5$, $N_P = 100$ and $G_{max} = 300$. That is because we have interest in the influence of the selection schemes. However, for two hard problems, i.e., DTLZ1 and DTLZ3 with $(M, D) = (5, 10)$, the maximum generation was set as $G_{max} = 400$. Then each MOEA was applied to each test problem 10 times.

Table 4 shows the result of the numerical experiment, namely the computational time ([ms]) evaluated by using

Table 4: Comparison of MOEAs

(a) DTLZ1 ($M = 2$, $D = 7$)

MOEA	[ms]	H	CM	FR
GDE3	329	1.250	0.000	1.001
IBDE	532	1.250	0.000	1.000
IBDE2	729	1.250	0.000	1.000
IBDE2R	8115	1.250	0.000	1.000

(b) DTLZ2 ($M = 2$, $D = 7$)

MOEA	[ms]	H	CM	FR
GDE3	386	1.500	0.000	2.000
IBDE	772	1.500	0.000	2.000
IBDE2	1673	1.500	0.000	2.000
IBDE2R	51164	1.500	0.000	2.000

(c) DTLZ3 ($M = 2$, $D = 7$)

MOEA	[ms]	H	CM	FR
GDE3	324	1.500	0.001	2.001
IBDE	509	1.500	0.002	2.002
IBDE2	611	1.501	0.003	2.003
IBDE2R	4161	1.500	0.001	2.001

(d) DTLZ4 ($M = 2$, $D = 7$)

MOEA	[ms]	H	CM	FR
GDE3	390	1.500	0.000	2.000
IBDE	779	1.500	0.000	2.000
IBDE2	1682	1.500	0.000	2.000
IBDE2R	51022	1.500	0.000	2.000

(e) DTLZ1 ($M = 5$, $D = 10$)

MOEA	[ms]	H	CM	FR
GDE3	759	1.258	8.032	124.7
IBDE	968414	1.256	0.009	2.374
IBDE2	8857	1.234	0.008	1.766
IBDE2R	339893	1.251	0.002	1.575

(f) DTLZ2 ($M = 5$, $D = 10$)

MOEA	[ms]	H	CM	FR
GDE3	657	1.518	0.010	5.117
IBDE	1096102	1.504	0.013	4.997
IBDE2	9440	1.509	0.023	4.950
IBDE2R	485690	1.500	0.000	5.000

(g) DTLZ3 ($M = 5$, $D = 10$)

MOEA	[ms]	H	CM	FR
GDE3	757	—	8772.	1017.
IBDE	700648	1.519	0.015	4.983
IBDE2	7956	1.510	0.022	4.949
IBDE2R	259456	1.533	0.003	5.066

(h) DTLZ4 ($M = 5$, $D = 10$)

MOEA	[ms]	H	CM	FR
GDE3	704	1.500	0.000	4.703
IBDE	656511	1.500	0.013	4.894
IBDE2	9136	1.494	0.018	4.890
IBDE2R	453100	1.500	0.000	4.999

Table 5: Example of ANOVA about CM

(a) DTLZ1 ($M = 2$, $D = 7$)

factor	ϕ	V	F	P	α
MOEA	3	$2.78E-8$	0.929	0.436	
error	36	$2.99E-8$			

(b) DTLZ1 ($M = 5$, $D = 10$)

factor	ϕ	V	F	P	α
MOEA	3	161.021	6.142	0.001	▽
error	36	26.213			

Table 6: Summary of multiple comparison test

$M = 5$	GDE3	IBDE	IBDE2	IBDE2R
DTLZ1		▽	▽	▽
DTLZ2				▽
DTLZ3		▽	▽	▽
DTLZ4	▽			▽

Table 7: Computational times of MOEAs

(a) DTLZ3 ($M = 2$, $D = 7$) [ms]

N_T	1	2	4	8
IBDE	531	468	481	509
IBDE2	884	668	618	611
IBDE2R	5534	3893	3693	4161

(b) DTLZ3 ($M = 5$, $D = 10$) [ms]

N_T	1	2	4	8
IBDE	1655125	1025675	736606	700648
IBDE2	17153	10559	8931	7956
IBDE2R	554415	313015	276334	259456

$N_T = 8$ threads except GDE3, the hypervolume (H), CM defined by (11) and (12), and Function Range (FR). FR is used to evaluate the diversity of $\boldsymbol{x} \in \Phi$ and defined as

$$FR(\boldsymbol{x}) = \sum_{m=1}^{M} [\max_{\boldsymbol{x} \in \Phi}\{f_m(\boldsymbol{x})\} - \min_{\boldsymbol{x} \in \Phi}\{f_m(\boldsymbol{x})\}] \qquad (13)$$

From the results in Table 4, in case of $M = 2$ objectives, there is not so much difference among the performances of MOEAs except the computational time. GDE3 is the fastest. IBDE2R spends the longest computational time among four MOEAs. On the other hand, in case of $M = 5$ objectives, the performances of MOEAs differ from each other depending on the test problems. Approximate fronts achieved by GDE3 have not converged to the Pareto fronts of DTLZ1 and DTLZ3 because the values of CM are large. Besides, we cannot evaluate the value of $H(\phi, \boldsymbol{r})$ for the solutions $\boldsymbol{x} \in \Phi$ of DTLZ3 obtained by GDE3. That is because some solutions $\boldsymbol{x} \in \Phi$ have exceeded a specified reference point $\boldsymbol{r} = (2, \cdots, 2) \in \mathbb{R}^M$. The performance of IBDE2R is the most excellent in all test problems except for the computational time. Furthermore, IBDE2R is faster than IBDE. In case of $M = 5$ objectives, IBDE spends the longest computational time among the four MOEAs.

The behavior of MOEA is always probabilistic. However, the run times of each MOEA have been limited in the above numerical experiments. Therefore, the results of the numerical experiments were verified statistically by using Analysis of Variance (ANOVA) [13]. The kind of MOEAs was selected as the factor. Consequently, the factor has four levels: GDE3, IBDE, IBDE2 and IBDE2R. By using ANOVA, the effect of the factor on the value of CM was evaluated through the eight test problems in Table 4. Table 5 shows two typical examples of ANOVA. In Table 5, degree of freedom (ϕ), variance (V), F-value, P-value and the risk (α) are listed. In case of DTLZ1 with $M = 2$ objectives, the difference of MOEAs doesn't have any effect on CM. However, in case of $M = 5$ objectives, it can be confirmed that the value of CM depends on the kind of MOEAs.

By using ANOVA, it could be verified that the difference of MOEAs influenced the value of CM in every test problem with $M = 5$ objectives. Therefore, in order to specify the most suitable MOEA for each test problem, the multiple comparison tests [13] were applied. Table 6 summarizes the results of the multiple comparison tests. For each of the test problems, symbol "$\bigtriangledown$" denotes the best MOEA with the risk less than 0.01. For example, in order to minimize CM of DTLZ1, the three variants of IBDE are more effective than GDE3. However, there is not a significant difference among the values of CM achieved by the three variants of IBDE. From the result of Table 6, IBDE2R is the most excellent at the value of CM in every test problem with $M = 5$ objectives. Even though the results of the statistical tests about H and FR are omitted for the restriction of pages, we have confirmed that IBDE2R is the most excellent at every criterion except the computational time.

Finally, the computational times of three variants of IBDE were evaluated through the eight test problems in Table 4. Table 7 shows two typical examples of the experimental results. In case of $M = 2$ objectives, the effect of the parallelized IHSO is small. However, in case of $M = 5$ objectives, the computational times of the three variants of IBDE decreased with the increase of the number of threads.

6. CONCLUSIONS

This paper proposed an indicator-based DE (IBDE). As an indicator function of IBDE, an exclusive hypervolume approximation ($EH2$) was also proposed. The proposed $EH2$ gave an upper bound of the exclusive hypervolume (EH). From the results of numerical experiments and statistical tests conducted on test problems, it was confirmed that there was not a significant difference between two evaluations of solutions based on EH and $EH2$. Besides, there was not so much difference between the performances of two variants of IBDE using EH and $EH2$ respectively except their computational times. Comparing $EH2$ with EH, an advantage of $EH2$ was a short computational time for the approximation set $\boldsymbol{x} \in \Phi$ with many objectives. The short computational time of $EH2$ enabled IBDE to employ the iterative selection scheme. By using the iterative selection with $EH2$, IBDE2R outperformed the other two variants of IBDE and GDE3 in the quality of the obtained approximation set Φ.

In our future works, we would like to apply the proposed IBDE2R to solve effectively the real-world application [24] formulated as a many-objective optimization problem.

7. REFERENCES

[1] J. Bader and E. Zizler. Hype: An algorithm for fast hypervolume-based many-objective optimization. In *TIK-Technical Report No. 286*, pages 1–25, 2008.

[2] L. Bradstreet, L. While, and L. Barone. Incrementally maximizing hypervolume for selection in multi-objective evolutionary algorithms. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2007)*, pages 3203–3210, 2007.

[3] L. Bradstreet, L. While, and L. Barone. A fast incremental hypervolume algorithm. *IEEE Trans. on Evolutionary Computation*, 12(6):714–723, 2008.

[4] C. Breshears. *The Art of Concurrency - A Thread Monkey's Guide to Writing Parallel Applications.* O'Reilly, 2009.

[5] K. Bringmann and T. Friedrich. Approximating the volume of unions and intersections of high-dimensional geometric objects. In *Proceedings of the 19th International Symposium on Algorithms and Computation (ISAAC)*, pages 436–447, 2008.

[6] K. Bringmann and T. Friedrich. Approximating the least hypervolume contributor: Np-hard in general, but fast in practice. In *Proceedings of the 5th International Conference on Evolutionary Multi-criterion Optimization (EMO'09)*, pages 6–20, 2009.

[7] D. Brockhoff and E. Zitzler. Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2007)*, pages 2086–2093, 2007.

[8] 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):182–197, 2002.

[9] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable test problems for evolutionary multi-objective optimization. In *TIK-Technical Report No. 112*, pages 1–27, 2001.

[10] M. Emmerich, N. Beume, and B. Naujoks. An emo algorithm using the hypervolume measure as selection criterion. In *Proceedings of the 3th International Conference on Evolutionary Multi-criterion Optimization (EMO'05)*, pages 62–76, 2005.

[11] V. Feoktistov. *Differential Evolution - In Search of Solutions.* Springer, 2006.

[12] C. M. Fonseca, L. Paquete, and M. Lopez-Ibanez. An improved dimension-sweep algorithm for the hypervolume indicator. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2006)*, pages 3973–3979, 2006.

[13] G. Gamst, L. S. Meyers, and A. J. Guarino. *Analysis of Variance Design: A Conceptual and Computational Approach with SPSS and SAS.* Cambridge University Press, 2008.

[14] E. J. Hughes. Evolutionary many-objective optimization: many once or one many? In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2005)*, pages 222–227, 2005.

[15] A. W. Iorio and X. Li. Solving rotated multi-objective optimization problems using differential evolution. In *Proceedings of the 17th Australian Joint Conference on Artificial Intelligence (AI2004)*, pages 861–872, 2004.

[16] A. W. Iorio and X. Li. Incorporating directional information within a differential evolution algorithm for multi-objective optimization. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO2006)*, pages 675–682, 2006.

[17] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima. Hypervolume approximation using achievement scalarizing functions for evolutionary many-objective optimization. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2009)*, pages 530–537, 2009.

[18] S. Kukkonen and J. Lampinen. An extension of generalized differential evolution for multi-objective optimization with constraints. In *Proceedings of Parallel Problem Solving from Nature (PPSN VIII)*, pages 752–761, 2004.

[19] S. Kukkonen and J. Lampinen. Gde3: The third evolutionary step of generalized differential evolution. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2005)*, pages 443–450, 2005.

[20] S. Kukkonen and J. Lampinen. Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2009)*, pages 1943–1950, 2009.

[21] K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution - A Practical Approach to Global Optimization*. Springer, 2005.

[22] R. Storn and K. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous space. *Journal of Global Optimization*, 11(4):341–359, 1997.

[23] K. Tagawa and T. Ishimizu. Concurrent implementation of differential evolution. In *Proceedings of the 10th WSEAS International Conference on System Theory and Scientific Computation (ISTASC'10)*, pages 65–70, 2010.

[24] K. Tagawa, Y. Sasaki, and H. Nakamura. Optimal design of balanced saw filters using multi-objective differential evolution. In *Proceedings of 8th International Conference on Simulated Evolution and Learning (SEAL'10)*, pages 466–475, 2010.

[25] S. Tsutsui and N. Fujimoto. Solving quadratic assignment problems by genetic algorithms with gpu computation: a case study. In *Proceedings of IEEE Congress on Evolutionary Computation (CEC2009)*, pages 2523–2530, 2009.

[26] T. Tusar and B. Filipic. Differential evolution versus genetic algorithms in multiobjective optimization. In *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization (EMO'7)*, pages 257–271, 2007.

[27] T. Wagner, N. Beume, and B. Naujoks. Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Proceedings of the 4th International Conference on Evolutionary Multi-criterion Optimization (EMO'7)*, pages 742–756, 2007.

[28] L. While, P. Hingston, L. Barone, and S. Huband. A faster algorithm for calculating hypervolume. *IEEE Trans. on Evolutionary Computation*, 10(1):29–38, 2006.

[29] E. Zitzler and S. Kunzli. Indicator-based selection in multiobjective search. In *Proceedings of Parallel Problem Solving from Nature (PPSN VIII)*, pages 832–842, 2004.

[30] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans. on Evolutionary Computation*, 3(4):257–271, 1999.

APPENDIX

A. PROOF OF THEOREM 1

PROOF. Let Φ be an approximation set of the problem in (5) and $\boldsymbol{x},\ \boldsymbol{z} \in \Phi$. According to (9), it holds:

$$\begin{aligned} &EH(\boldsymbol{x}, \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) = H(\{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) - H(\{\boldsymbol{z}\}, \boldsymbol{r}) \\ &= vol(B(\{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r})) - vol(B(\{\boldsymbol{z}\}, \boldsymbol{r})) \end{aligned} \tag{14}$$

By dividing $\{\boldsymbol{x}, \boldsymbol{z}\}$ into two groups:

$$B(\{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) = B(\{\boldsymbol{x}\}, \boldsymbol{r}) \cup B(\{\boldsymbol{z}\}, \boldsymbol{r}) \supseteq B(\{\boldsymbol{z}\}, \boldsymbol{r}) \tag{15}$$

From (15), (14) can be rewritten as

$$\begin{aligned} &EH(\boldsymbol{x}, \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) = H(\{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) - H(\{\boldsymbol{z}\}, \boldsymbol{r}) \\ &= vol(B(\{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) - B(\{\boldsymbol{z}\}, \boldsymbol{r})) \\ &= vol(B(\{\boldsymbol{x}\}, \boldsymbol{r}) - B(\{\boldsymbol{x}\}, \boldsymbol{r}) \cap B(\{\boldsymbol{z}\}, \boldsymbol{r})) \end{aligned} \tag{16}$$

According to (9), it holds:

$$\begin{aligned} &EH(\boldsymbol{x}, \Phi, \boldsymbol{r}) = H(\Phi, \boldsymbol{r}) - H(\Phi \setminus \boldsymbol{x}, \boldsymbol{r}) \\ &= vol(B(\Phi, \boldsymbol{r})) - vol(B(\Phi \setminus \boldsymbol{x}, \boldsymbol{r})) \end{aligned} \tag{17}$$

By dividing Φ into two groups:

$$B(\Phi, \boldsymbol{r}) = B(\Phi \setminus \boldsymbol{x}, \boldsymbol{r}) \cup B(\{\boldsymbol{x}\}, \boldsymbol{r}) \supseteq B(\Phi \setminus \boldsymbol{x}, \boldsymbol{r}) \tag{18}$$

From (18), (17) can be rewritten as

$$\begin{aligned} &EH(\boldsymbol{x}, \Phi, \boldsymbol{r}) = vol(B(\Phi, \boldsymbol{r}) - B(\Phi \setminus \boldsymbol{x}, \boldsymbol{r})) \\ &= vol(B(\{\boldsymbol{x}\}, \boldsymbol{r}) \cup B(\{\boldsymbol{z}\}, \boldsymbol{r}) \cup B(\Phi \setminus \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) \\ &\quad -B(\{\boldsymbol{z}\}, \boldsymbol{r}) \cup B(\Phi \setminus \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r})) \\ &= vol(B(\{\boldsymbol{x}\}, \boldsymbol{r}) - B(\{\boldsymbol{x}\}, \boldsymbol{r}) \cap B(\{\boldsymbol{z}\}, \boldsymbol{r}) \\ &\quad -(B(\{\boldsymbol{x}\}, \boldsymbol{r}) - B(\{\boldsymbol{x}\}, \boldsymbol{r}) \cap B(\{\boldsymbol{z}\}, \boldsymbol{r})) \\ &\quad \cap B(\Phi \setminus \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r})) \end{aligned} \tag{19}$$

From the set theory, it holds:

$$\begin{aligned} &B(\{\boldsymbol{x}\}, \boldsymbol{r}) - B(\{\boldsymbol{x}\}, \boldsymbol{r}) \cap B(\{\boldsymbol{z}\}, \boldsymbol{r}) \supseteq (B(\{\boldsymbol{x}\}, \boldsymbol{r}) \\ &\quad -B(\{\boldsymbol{x}\}, \boldsymbol{r}) \cap B(\{\boldsymbol{z}\}, \boldsymbol{r})) \cap B(\Phi \setminus \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) \end{aligned} \tag{20}$$

From (20), (19) can be rewritten as

$$\begin{aligned} &EH(\boldsymbol{x}, \Phi, \boldsymbol{r}) = vol(B(\{\boldsymbol{x}\}, \boldsymbol{r}) \\ &\quad -B(\{\boldsymbol{x}\}, \boldsymbol{r}) \cap B(\{\boldsymbol{z}\}, \boldsymbol{r})) - vol((B(\{\boldsymbol{x}\}, \boldsymbol{r}) \\ &\quad -B(\{\boldsymbol{x}\}, \boldsymbol{r}) \cap B(\{\boldsymbol{z}\}, \boldsymbol{r})) \cap B(\Phi \setminus \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r})) \end{aligned} \tag{21}$$

From (16) and (21), it holds:

$$EH(\boldsymbol{x}, \{\boldsymbol{x}, \boldsymbol{z}\}, \boldsymbol{r}) \geq EH(\boldsymbol{x}, \Phi, \boldsymbol{r}) \tag{22}$$

Since (22) holds for $\forall \boldsymbol{x},\ \boldsymbol{z} \in \Phi$, we obtain

$$EH2(\boldsymbol{x}, \Phi, \boldsymbol{r}) \geq EH(\boldsymbol{x}, \Phi, \boldsymbol{r}) \tag{23}$$

which concludes the proof. □