A grid-based evolutionary algorithm for many-objective optimization

Yang, Shengxiang; Li, Miqing; Liu, Xiaohui; Zheng, Jinhua

Citation for published version (Harvard):
Yang, S, Li, M, Liu, X & Zheng, J 2013, 'A grid-based evolutionary algorithm for many-objective optimization', *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5.

A Grid-Based Evolutionary Algorithm for Many-Objective Optimization

Shengxiang Yang, *Member, IEEE*, Miqing Li, Xiaohui Liu, and Jinhua Zheng

***Abstract*—Balancing convergence and diversity plays a key role in evolutionary multiobjective optimization (EMO). Most current EMO algorithms perform well on problems with two or three objectives, but encounter difficulties in their scalability to many-objective optimization. This paper proposes a Grid-based Evolutionary Algorithm (GrEA) to solve many-objective optimization problems. Our aim is to exploit the potential of the grid-based approach to strengthen the selection pressure towards the optimal direction while maintaining an extensive and uniform distribution among solutions. To this end, two concepts—*grid dominance* and *grid difference*—are introduced to determine the mutual relationship of individuals in a grid environment. Three grid-based criteria, i.e., *grid ranking*, *grid crowding distance*, and *grid coordinate point distance*, are incorporated into the fitness of individuals to distinguish them in both the mating and environmental selection processes. Moreover, a fitness adjustment strategy is developed by adaptively punishing individuals based on the neighborhood and grid dominance relations in order to avoid partial overcrowding as well as guide the search towards different directions in the archive. Six state-of-the-art EMO algorithms are selected as the peer algorithms to validate GrEA. A series of extensive experiments is conducted on 52 instances of 9 test problems taken from 3 test suites. The experimental results show the effectiveness and competitiveness of the proposed GrEA in balancing convergence and diversity. The solution set obtained by GrEA can achieve a better coverage of the Pareto front than that obtained by other algorithms on most of the tested problems. Additionally, a parametric study reveals interesting insights of the division parameter in grid and also indicates useful values for problems with different characteristics.**

***Index Terms*—Evolutionary multiobjective optimization, many-objective optimization, convergence, diversity, grid.**

I. Introduction

In the real world, it is not uncommon to face a problem with several objectives to be met simultaneously, which is known as a multiobjective optimization problem (MOP). Due to the conflicting nature of objectives, there is usually no single optimal solution but rather a set of alternative solutions, called the Pareto set, for MOPs. Evolutionary algorithms (EAs) have been recognized to be well suitable for MOPs due to their population-based property of achieving an approximation of the Pareto set in a single run. Over the past few decades, a number of state-of-the-art evolutionary multiobjective optimization (EMO) algorithms have been proposed. Generally speaking, these algorithms share two common but often conflicting goals—minimizing the distance of solutions to the optimal front (i.e., convergence) and maximizing the distribution of solutions over the optimal front (i.e., diversity).

A many-objective optimization problem usually refers to an optimization problem with more than three objectives. It appears widely in industrial and engineering design, such as water resource engineering [38], industrial scheduling problem [64], control system design [19], [23], molecular design [44], and so on [18], [31], [36]. In recent years, many-objective optimization has been gaining an increasing attention in the EMO community [60], [63]. Some related techniques have been developed rapidly in the domain, including test functions scalable to any number of conflicting objectives [16], [24], [25], [61], performance assessment metrics suitable for a high-dimensional space [14], [34], and visualization tools designed for the display of solutions with four or more objectives [33], [43], [55], [67]. These have made it possible to deeply investigate the performance of algorithms on many-objective problems. As a consequence, various experimental [9], [21], [32] and analytical [8], [62], [65] studies have been presented and some new observations and conclusions have been made in the many-objective optimization landscape [18], [42], [56].

Balancing convergence and diversity is not an easy task in many-objective optimization. Most classical Pareto-based EMO algorithms, such as the improved nondominance sorting genetic algorithm (NSGA-II) [11] and the improved strength Pareto EA (SPEA2) [73], noticeably deteriorate their search ability when more than three objectives are involved [39], [66]. One major reason for this occurrence is that the proportion of nondominated solutions in a population rises rapidly with the number of objectives. This makes the Pareto-based primary selection fail to distinguish individuals and the secondary selection based on diversity play a leading role in determining the survival of individuals. In this case, the performance of algorithms may worsen since they prefer *dominance resistant solutions* [29] (i.e., the solutions with an extremely poor value in at least one of the objectives, but with near optimal values in the others). Consequently, the solutions in the final solution set may be distributed uniformly in the objective space, but away from the desired Pareto front. In fact, some studies have shown that a random search algorithm may even achieve better results than Pareto-based algorithms in MOPs with a high number of objectives [42], [45], [56].

A straightforward idea to solve this problem (i.e., the ineffectiveness of classical Pareto-based EMO algorithms in many-objective optimization) is to modify or enhance the Pareto dominance relation to increase the selection pressure towards the Pareto front. There are a large number of studies

S. Yang is with the School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U. K. (e-mail: syang@dmu.ac.uk).

M. Li and X. Liu are with the Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex UB8 3PH, U. K. (email: {miqing.li, xiaohui.liu}@brunel.ac.uk).

J. Zheng is with the College of Information Engineering, Xiangtan University, Xiangtan, China (e-mail: jhzheng@xtu.edu.cn).

that address this issue in different ways, e.g., dominance area control [6], [59], *k*-optimality [18], preference order ranking [17], subspace dominance comparison [2], [35], ϵ-dominance [46], and fuzzy Pareto dominance [43]. Experimental results indicate that the selection based on these modified Pareto dominance relation is able to search towards the Pareto front significantly better than that based on the standard Pareto dominance relation. On the other hand, some non-Pareto-based techniques, such as average rank [49], ranking dominance [45], favour relation [64], winning score [51], and some distance-based rank methods [20], [53], [69], [75], have also been found to be promising in converging towards the optimum. Despite the risk of leading the final set to converge into a sub-area in the Pareto front [9], [34], they provide some new alternatives for evolutionary many-objective optimization.

Another avenue to deal with this problem is to improve the diversity maintenance mechanism of the algorithms. As a result of the increase of the objective space in size, the conflict between the convergence and diversity requirements is gradually aggravated [1], [56]. Most of the diversity maintenance techniques (e.g., niche, crowding distance, clustering, and *k*-th nearest distance [12]) not only cannot strengthen the selection pressure towards the Pareto front, but also hinder the evolutionary search to some extent due to their favour on dominance resistant solutions [56]. Apparently, a feasible way for solving this problem is to decrease the diversity requirement in the selection process. Some efforts in this direction have been attempted. Adra and Fleming [1] employed a diversity management operator to control the diversity requirement in order to maintain the balance between the two requirements. Wagner *et al.* [66] demonstrated that the assignment of a zero distance (instead of an infinity distance) to boundary solutions would clearly improve the performance of NSGA-II for many-objective problems.

Recently, some selection methods which naturally integrate proximity and diversity into a single criterion have been developed, e.g., the aggregation-based criterion [26], [28], [71] and indicator-based criterion [3], [4], [74]. Some EMO algorithms based on them, e.g., MSOPS [26] and SMS-EMOA [4], have been found to perform well in balancing convergence and diversity for many-objective problems [66]. The former uses the idea of single-objective aggregated optimization to search in parallel for points that lie on the Pareto front. The latter aims at maximizing the hypervolume contribution of the population by selecting the individuals in a steady state evolutionary scheme. While a large computation cost is required in the exact calculation of the hypervolume indicator in a high-dimensional space, efforts [7], [68] to address this issue are being made; on the other hand, approximate estimation of the hypervolume indicator has recently been developed by Monte Carlo sampling, leading to the appearance of hypervolume-based algorithms designed specially for many-objective optimization, such as HypE [3].

The above studies provide a variety of alternatives to address many-objective optimization problems and many of them offer bright prospects to the domain. Nevertheless, great improvements are still needed before EMO algorithms can be considered as an effective tool for many-objective problems as for 2- or 3-objective problems. As highlighted by Purshouse *et al.* in [57], research into evolutionary many-objective optimization is still in its infancy, and the need for efficient methodologies is pressing.

This paper proposes a grid-based EA (GrEA) to solve many-objective optimization problems. The aim of the paper is to exploit the potential of the grid-based approach to strengthen the selection pressure towards the optimal direction while maintaining an extensive and uniform distribution among solutions.

Grid has an inherent property of reflecting the information of convergence and diversity simultaneously. Each solution in the grid has a deterministic location. The performance of a solution regarding convergence can be estimated by its grid location compared with other solutions, and the performance of a solution regarding diversity can be estimated by the number of solutions whose grid locations are identical with or similar to its grid location. Moreover, in contrast to the Pareto dominance criterion, a grid-based criterion can not only qualitatively compare solutions but also give the quantitative difference in each objective among them. This seems to be more suitable for many-objective problems, considering the increase of selection pressure from the quantitative comparison of objective values among solutions [9], [31]. However, so far the advantage of grid is not utilized adequately in the literature (see Section II for an overview of existing studies). Most existing grid-based EMO algorithms perform well on problems with two or three objectives, but often fail to deal with many-objective problems. In addition, some aggregation-based algorithms that have been found to be successful in both multi- and many-objective problems, such as cellular EAs [54] and the multiobjective EA based on decomposition (MOEA/D) [71], also employ grid-structure to guide the search. However, an essential difference from the grid-based approaches here is that they decompose an MOP into many single-objective problems by a set of weight vectors with grid structure, and steer the search towards the Pareto front in different directions according to these uniformly-distributed weight vectors.

As the first attempt to capture and utilize the properties of grid for evolutionary many-objective optimization, we recently developed a grid-based fitness strategy [50]. Three fitness assignment criteria based on individuals' grid coordinates are used to reinforce the selection pressure towards the optimal front. The comparison studies in [50] has shown that the proposed strategy is competitive with several EMO algorithms on the tested many-objective problems. Motivated by the encouraging experimental results of the preliminary attempt, this paper conducts a further and thorough investigation along this line and proposes the GrEA to tackle many-objective optimization problems.

The main contributions of GrEA distinguished from its predecessor can be summarized as follows.

- A concept of *grid dominance* is introduced to compare individuals in both the mating and environmental selection processes.
- An elaborate density estimator of an individual in the population is designed, which takes into account not only the number of its neighbors but also the distance

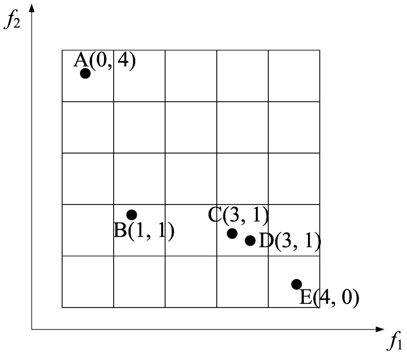

Fig. 1. An illustration of individuals in grid in a bi-objective space.

difference between itself and these neighbors.

- An improved fitness adjustment technique is developed to avoid partial overcrowding as well as guide the search towards different directions in the archive set.

In this paper, systematic experiments are carried out to compare GrEA with five other state-of-the-art algorithms on three groups of many-objective continuous and combinatorial optimization problems. In addition, a detailed investigation of the grid division parameter is also included in our study.

The remainder of this paper is organized as follows. In Section II, the motivation of using grid to solve many-objective problems and the related work concerning grid-based techniques are delineated and discussed. Section III is devoted to the description of the proposed algorithm for many-objective optimization problems. Section IV presents the algorithm settings, test functions, and performance metrics used for performance comparison. The experimental results and discussions are given in Section V. Finally, Section VI provides some concluding remarks along with pertinent observations.

II. Motivation and Related Work

Grid has a natural ability to reflect the distribution of solutions in the evolutionary process by their own grid locations (i.e., grid coordinates). The difference between grid coordinates of solutions indicates the distance between solutions and further images the density information of solutions in the population. For example, Fig. 1 illustrates individuals in grid in a bi-objective space. For individuals **A**, **B**, and **C** in the figure, their grid coordinates are $(0, 4)$, $(1, 1)$, and $(3, 1)$, respectively. Clearly, the difference of grid coordinates between **A** and **C** (i.e., $(3-0)+(4-1)=6$) is larger than that between **A** and **B** (i.e., $(1-0)+(4-1)=4$), which indicates that **C** is farther away from **A** than **B**. In addition, given that there exists another individual (**D**) which has the identical grid coordinate with **C** (i.e., the difference of grid coordinates between them is 0), **C** can be considered to be of a greater crowding degree in comparison with **A** and **B**.

On the other hand, grid is also capable of indicating the evolutionary status of solutions in terms of convergence. The grid coordinate takes into account not only whether one solution is better than another solution but also the difference in objective values between them. For example, considering individuals **A** and **B** with their own grid coordinates $(0, 4)$ and $(1, 1)$ in Fig. 1, it is clear that the difference in objective f_2 between them is greater than that in objective f_1 (i.e., $(4-1) > (1-0)$). This means that grid can further distinguish the solutions when they are on a tie in the sense of Pareto dominance, thereby providing a higher selection pressure in the evolutionary many-objective optimization process.

Over the last decade, grid-based techniques are widely applied in the EMO community, resulting in the appearance of several grid-based EMO algorithms. Their characteristics have also been well studied, both theoretically and experimentally [15], [41], [46], [58], [72]. In the first known study of this kind, grid was introduced into Pareto-based evolution strategy (PAES) proposed by Knowles and Corne [40] to maintain the diversity of the archive set. The crowding degree of a solution is estimated by the number of solutions sharing its grid location. When a nondominated candidate is to join an archive that is full, it replaces one of the solutions with the highest crowding degree if its own crowding degree is lower. Some extended theoretical and practical studies were also presented in [10], [41].

Yen and Lu [70] presented a dynamic multiobjective EA (DMOEA), using adaptive grid-based rank and density estimation. Unlike PAES and PESA, grid, here, is regarded as an implement to store the information of both convergence and diversity of solutions. Each cell in grid is assigned a rank and a density value according to the Pareto dominance relation and grid location of solutions.

The concept of ϵ-dominance proposed firstly by Laumanns *et al.* [46] can be seen as a grid-based technique to combine the convergence properties of an EMO algorithm with the need to preserve a diverse set of solutions. Deb *et al.* [15] also developed a steady-state algorithm ϵ-MOEA using ϵ-dominance. It divides the objective space into hyperboxes by the size of ϵ and each hyperbox contains at most a single individual. However, the boundary solutions may be lost in the evolutionary process of the algorithm due to the feature of ϵ-dominance [15], [22]. To address this issue, Hernández-Díaz *et al.* [22] proposed a variant of the algorithm, called Pareto adaptive ϵ-dominance.

Rachmawati and Srinivasan [58] introduced a dynamic grid resizing strategy, which is capable of shrinking or expanding hyperboxes as necessity dictates. The strategy uses two metrics—the mean occupancy and neighbor occupancy—to detect the setting of grid sizes, and further adjusts them correspondingly.

More recently, Karahan and Köksalan [37] developed a territory-based EMO algorithm, TDEA, to solve MOPs. Similar to ϵ-MOEA, TDEA is also a steady-state algorithm. It defines a territory τ around an individual to maintain diversity. Its main difference from ϵ-MOEA lies in that the hyperbox of TDEA is based on individuals rather than independent on them.

Overall, the above grid-based EMO algorithms are very successful, and most of them perform very well on the problems with two or three objectives. However, it is interesting that their application to many-objective problems has received little attention and consideration. This occurrence may mainly be attributed to three reasons, summarized as follows.

- The need of data storage and computational time increases exponentially. The calculation of most existing grid-based algorithms revolves around hyperboxes in grid. Such box-centered calculation often needs to store the information of each hyperbox in grid (e.g., the number of individuals in each hyperbox). As pointed out by Corne and Knowles [9], these algorithms may not be suitable for many-objective problems since their operation relies on the data structures that exponentially grow in size with the number of objectives. Additionally, the computational cost for high dimensional problems would also be tremendous when the box-centered calculation is implemented [39]. If we traverse each hyperbox in an m-dimensional grid, there will be r^m hyperboxes to be accessed, where r is the number of divisions in each dimension.
- The properties of grid are not utilized or exploited sufficiently. The selection criterion of some grid-based EMO algorithms (such as PAES, DMOEA, and TDEA), in the sense of convergence, is based on the Pareto dominance relation, and thus may fail to provide enough selection pressure towards the desired direction in the evolutionary process of many-objective optimization.
- The density estimator may fail to reflect the distribution of solutions accurately. Since the number of hyperboxes in grid exponentially increases with the number of objectives, the solutions for many-objective problems are likely to disperse in different hyperboxes. Consequently, existing grid-based EMO algorithms which only consider the number of individuals in a single hyperbox would not discriminate individuals by means of their distribution, as the values are almost equal based on this density estimation method.

Clearly, the above difficulties would largely limit the application of existing grid-based EMO algorithms to many-objective problems. However, we argue that these difficulties do not seem to be insurmountable. Firstly, box-centered calculation can be replaced by individual-centered calculation. In this case, grid is merely regarded as a "pointer" to depict the address of individuals. Secondly, a selection criterion based on the difference of grid coordinates can be introduced to strengthen the selection pressure. Finally, the failure of density estimation may also be addressed if the density value of individuals relies on the records not in a single hyperbox but rather in a region constructed by a set of hyperboxes whose range increases with the number of objectives.

Bearing these ideas and motivations in mind, a grid-based evolutionary many-objective optimization algorithm is suggested, investigated, and discussed in the following sections.

III. Development of the Proposed Algorithm

In this section, we first introduce some definitions used in GrEA. Then, we present the framework of the proposed algorithm. Next, we describe the fitness assignment process. Finally, the strategies for mating and environmental selection processes are given in Section III-D and Section III-E, respectively.

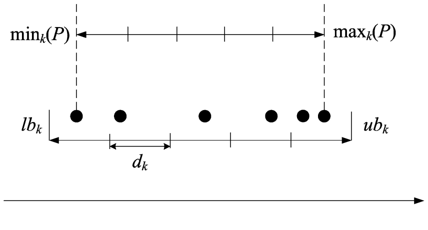

Fig. 2. Setting of grid in the kth objective.

A. Definitions and Concepts

Without loss of generality, an MOP may be stated as a minimization problem and defined as follows:

$$\begin{array}{ll}\text{Minimize} & \boldsymbol{F}(\boldsymbol{x}) = (F_1(\boldsymbol{x}), F_2(\boldsymbol{x}), ..., F_M(\boldsymbol{x})) \\ \text{Subject to} & \boldsymbol{x} \in \boldsymbol{\Omega}\end{array} \tag{1}$$

where $\boldsymbol{x}$ denotes a solution vector in the feasible solution space $\boldsymbol{\Omega}$, and F_i $(i = 1, 2, ..., M)$ is the ith objective to be minimized.

Definition (Pareto Dominance): Let $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{\Omega}, \boldsymbol{x} \prec \boldsymbol{y}$: $\Leftrightarrow$

$$\begin{array}{l}\forall i \in (1, 2, ..., M) : F_i(\boldsymbol{x}) \leq F_i(\boldsymbol{y}) \wedge \\ \exists j \in (1, 2, ..., M) : F_j(\boldsymbol{x}) < F_j(\boldsymbol{y})\end{array} \tag{2}$$

where "$\boldsymbol{x} \prec \boldsymbol{y}$" denotes that "$\boldsymbol{x}$ dominates $\boldsymbol{y}$". The Pareto dominance relation reflects the weakest assumption about the preferred structure of the decision maker. A decision vector that is not dominated by any other vectors is denoted as *Pareto optimal*. The set of the optimal solutions in the decision space is denoted as the *Pareto set*, and the corresponding set of objective vectors is denoted as the *Pareto front*.

In GrEA, grid is used as a frame to determine the location of individuals in the objective space. Therefore, its adaptability with the evolutionary population seems to be advisable. In other words, when a new population is generated, the location and size of grid should be adapted and adjustable so that it just envelops the population. Here, we adopt the adaptive construction of grid, borrowing ideas from the adaptive genetic algorithm (AGA) presented by Knowles and Corne [41].

The grid setting in the kth objective is shown in Fig. 2. First, the minimum and maximum values regarding the kth objective among the individuals in a population P are found and denoted as $min_k(P)$ and $max_k(P)$, respectively. Then, the lower and upper boundaries of the grid in the kth objective are determined according to the following formulas:

$$lb_k = \min_k(P) - (\max_k(P) - \min_k(P))/(2 \times div) \tag{3}$$

$$ub_k = \max_k(P) + (\max_k(P) - \min_k(P))/(2 \times div) \tag{4}$$

where div denotes the number of the divisions of the objective space in each dimension (e.g., in Fig. 2, $div = 5$). Accordingly, the original M-dimensional objective space will be divided into div^M hyperboxes. Thus, the hyperbox width d_k in the kth objective can be formed as follows:

$$d_k = (ub_k - lb_k)/div \tag{5}$$

In this case, the grid location of an individual in the kth objective can be determined by lb_k and d_k:

$$G_k(\boldsymbol{x}) = \lfloor (F_k(\boldsymbol{x}) - lb_k)/d_k \rfloor \tag{6}$$

where "$\lfloor \cdot \rfloor$" denotes the floor function, $G_k(\boldsymbol{x})$ is the grid coordinate of individual $\boldsymbol{x}$ in the kth objective, and $F_k(\boldsymbol{x})$ is the actual objective value in the kth objective. For example, in Fig. 2, the grid coordinate of individuals (from left to right) in the kth objective is 0, 1, 2, 3, 4, and 4. In the following, two concepts used in the comparison between individuals are defined based on their grid coordinates.

Definition (Grid Dominance): Let $\boldsymbol{x}, \boldsymbol{y} \in P, \boldsymbol{x} \prec_{grid} \boldsymbol{y}$: $\Leftrightarrow$

$$\begin{aligned} &\forall i \in (1, 2, ..., M) : G_i(\boldsymbol{x}) \leq G_i(\boldsymbol{y}) \wedge \\ &\exists j \in (1, 2, ..., M) : G_j(\boldsymbol{x}) < G_j(\boldsymbol{y}) \end{aligned} \tag{7}$$

where "$\boldsymbol{x} \prec_{grid} \boldsymbol{y}$" denotes that "$\boldsymbol{x}$ grid-dominates $\boldsymbol{y}$", M is the number of objectives, and the grid environment is constructed by the population P. Apparently, the concept of grid dominance is the same as that of Pareto dominance if the grid coordinates of individuals are replaced by their actual objective values. Their specific relationship is as follows. If one solution Pareto-dominates another solution, the latter will not grid-dominate the former, and vice versa. On the other hand, the grid dominance relation permits one solution to dominate another solution if the former is slightly inferior to the latter in some objectives but largely superior to the latter in some other objectives, e.g., the individuals **B** and **C** in Fig. 1.

The grid dominance relation is also similar to the ϵ-dominance relation, considering that both of them are the relaxed form of the Pareto dominance relation. But, one important difference is that the degree of relaxation of grid dominance is determined by the evolutionary status of the population. The division number div in GrEA is a fixed parameter set by the user beforehand, leading the convergence and diversity requirements to be adjusted adaptively with the evolution of the population. A widely-distributed population in the objective space (often appearing at the initial stage of evolution) has a larger relaxation degree (i.e., a larger size of a cell in the grid), thereby providing a higher selection pressure; as the population evolves toward the more concentrated Pareto front region, the relaxation degree becomes lower, leading the diversity to be more emphasized.

In addition, the usage of grid dominance in our study is totally different from that of ϵ-dominance in the ϵ-dominance-based algorithms. In the ϵ-dominance-based algorithms, ϵ-dominance is used to determine the survival of individuals. Only nondominated individuals can be preserved in the archive set. However, in GrEA, grid dominance is mainly used to prevent individuals from being archived earlier than their competitors that grid-dominate them (see the fitness adjustment strategy in Section III-E). This means that grid-dominated individuals can also have the chance to enter the archive set, which is useful for the maintenance of boundary solutions in the population to some extent.

Definition (Grid Difference): Let $\boldsymbol{x}, \boldsymbol{y} \in P$, the grid difference between them is denoted as:

$$GD(\boldsymbol{x}, \boldsymbol{y}) = \sum_{k=1}^{M} \mid G_k(\boldsymbol{x}) - G_k(\boldsymbol{y}) \mid \tag{8}$$

Grid difference is influenced by the number of divisions div, ranging from 0 to $M(div-1)$. The larger the div, the smaller the size of a cell and the higher the grid difference value between individuals.

Algorithm 1 Grid-Based Evolutionary Algorithm (GrEA)

```
Require: $P$ (population), $N$ (population size)
$P \leftarrow Initialize(P)$
while termination criterion not fulfilled do
    $Grid\_setting(P)$
    $Fitness\_assignment(P)$
    $P' \leftarrow Mating\_selection(P)$
    $P'' \leftarrow Variation(P')$
    $P \leftarrow Environmental\_selection(P \bigcup P'')$
end while
return $P$
```

B. Framework of the Proposed Algorithm

Algorithm 1 gives the framework of GrEA. The basic procedure of the algorithm is similar to most generational EMO algorithms like NSGA-II [11] and SPEA2 [73]. Firstly, N individuals are randomly generated to form an initial population P. Then, the grid environment for the current population P is set as described in the previous section, and the fitness of individuals in P is assigned according to their location in the grid. Next, mating selection is performed to pick out promising solutions for variation. Finally, the environmental selection procedure is implemented to keep a record of the N best solutions (elitists) for survival.

C. Fitness Assignment

In order to evolve the population towards the optimum as well as diversify the individuals uniformly along the obtained trade-off surface, the fitness of individuals should contain the information in terms of both convergence and diversity. This paper takes three grid-based criteria into account to assign the fitness of individuals. They are grid ranking (*GR*), grid crowding distance (*GCD*), and grid coordinate point distance (*GCPD*). The first and last criteria are used to evaluate the convergence of individuals while the middle one is concerned with the diversity of individuals in the population.

GR is a convergence estimator to rank individuals based on their grid locations. For each individual, *GR* is defined as the summation of its grid coordinate in each objective:

$$GR(\boldsymbol{x}) = \sum_{k=1}^{M} G_k(\boldsymbol{x}) \tag{9}$$

where $G_k(\boldsymbol{x})$ denotes the grid coordinate of individual $\boldsymbol{x}$ in the kth objective, and M is the number of objectives.

GR can be considered as a natural tradeoff between the number of objectives for which one solution is better than another and the difference of values in a single objective between two

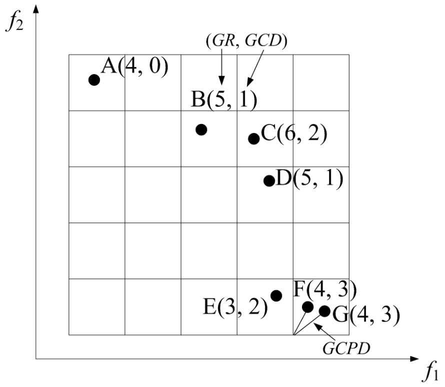

Fig. 3. Illustration of fitness assignment. The numbers in the brackets associated with each solution correspond to *GR* and *GCD*, respectively.

solutions. On the one hand, if an individual performs better than its competitors in the majority of objectives, it would have a higher likelihood of obtaining a lower *GR* value. On the other hand, the difference in a single objective is also an important part of influencing the *GR* value. For instance, considering individuals **C** and **A** in Fig. 3, **C** will obtain a worse *GR* value than **A** (6 against 4) since the advantage in f_2 is less than the disadvantage in f_1.

Note that the behavior of *GR* is closely related to the shape of the Pareto front of a multiobjective problem; e.g., individuals around the center of the Pareto front have good evaluations when the shape is convex, and individuals located in the edges of the Pareto front are preferable when the shape is concave. This may drive the population towards a certain area of the Pareto front, like the knee of the Pareto front. In our study, a *GR* adjustment strategy will be introduced to deal with this issue in the environmental selection process.

Density estimation of solutions is an important item in the fitness assignment process since a set of well-distributed solutions will play a crucial role in driving the search towards the entire Pareto front. However, the existing grid-based density estimators, which record the number of solutions occupying a single hyperbox, may fail to reveal their distribution because of the exponential increase of the number of hyperboxes with the number of objectives. Here, we enlarge the range of regions being considered and introduce the concept of the neighbors of a solution. A solution $\boldsymbol{y}$ is regarded as a neighbor of a solution $\boldsymbol{x}$ if $GD(\boldsymbol{x},\boldsymbol{y}) < M$, where $GD(\boldsymbol{x},\boldsymbol{y})$ denotes the *grid difference* between $\boldsymbol{x}$ and $\boldsymbol{y}$, and M is the number of objectives. GrEA considers the distribution of neighbors of a solution with respect to its density estimation. Specifically, the density estimator—grid crowding distance (*GCD*)—of $\boldsymbol{x}$ is defined as:

$$GCD(\boldsymbol{x}) = \sum_{\boldsymbol{y} \in N(\boldsymbol{x})} (M - GD(\boldsymbol{x},\boldsymbol{y})) \tag{10}$$

where $N(\boldsymbol{x})$ stands for the set of neighbors of $\boldsymbol{x}$. For instance, in Fig. 3, the neighbors of individual **G** are **E** and **F**, and the *GCD* of **G** is 3, i.e., $(2-1)+(2-0)=3$.

Clearly, the *GCD* of a solution depends on both the range of neighborhood (i.e., the region in which other solutions are regarded as its neighbors) and the grid difference between it and other solutions. On the one hand, a larger neighborhood range generally contains more solutions, thus contributing to a higher GCD value. Note that the neighborhood range is determined by M. The number of considered hyperboxes will gradually increase with the number of objectives, which would be consistent with the total number of hyperboxes in the grid environment, hence providing a clear distinction of the crowding degree among individuals. On the other hand, GCD also indicates the position information of solutions in the neighborhood since the grid difference metric is involved. The farther the neighbors are located, the smaller the contribution to GCD is. For example, considering individuals **C** and **F** in Fig. 3, the *GCD* of **C** is smaller than that of **F** (2 against 3) although the number of their neighbors is exactly equal.

Although *GR* and *GCD* have already provided a good measure of individuals in terms of convergence and diversity, they may still fail to discriminate individuals. Since their calculation is based on the grid coordinates of individuals, both *GR* and *GCD* have an integral value, which means that some individuals may have the same *GR* and *GCD* values, e.g., individuals **B** and **D** in Fig. 3. Here, inspired by the strategy in ϵ-MOEA [15], we calculate the normalized Euclidean distance between an individual and the utopia point in its hyperbox (i.e., the best corner of its hyperbox), called grid coordinate point distance (*GCPD*), as follows:

$$GCPD(\boldsymbol{x}) = \sqrt{\sum_{k=1}^{M} (\,(F_k(\boldsymbol{x}) - (lb_k + G_k(\boldsymbol{x}) \times d_k))/d_k\,)^2} \tag{11}$$

where $G_k(\boldsymbol{x})$ and $F_k(\boldsymbol{x})$ denote the grid coordinate and actual objective value of individual $\boldsymbol{x}$ respectively in the kth objective, lb_k and d_k stand for the lower boundary of the grid and the width of a hyperbox, respectively, for the kth objective, and M is the number of objectives. Clearly, a lower *GCPD* is preferable. Individuals **F** and **G** in Fig. 3 also illustrate this criterion.

According to the three grid-based criteria *GR*, *GCD*, and *GCPD*, the evolutionary status of individuals can be reflected effectively. In the following, we will employ these criteria to compare individuals in the selection process.

D. Mating Selection

Mating selection which aims to make a good preparation for exchanging the information of individuals plays an important role in EMO algorithms. It is usually implemented by selecting promising solutions from the current population to form a mating pool. Here, we use a type of binary tournament selection strategy based on the dominance relation and density information to pick out individuals for variation.

Algorithm 2 gives a detailed procedure of this strategy. First, two individuals are randomly chosen from the population. If one Pareto-dominates or grid-dominates the other, the former is chosen. Otherwise, it indicates that these two solutions are non-dominated to each other regarding both the Pareto dominance and grid dominance relations. In this case, we prefer the solution with a lower density estimation value (i.e.,

Algorithm 2 $TournamentSelection$

```
Require: individuals $p$, $q$ randomly chosen from the population
if $p \prec q$ or $p \prec_{grid} q$ then
    return $p$
else if $q \prec p$ or $q \prec_{grid} p$ then
    return $q$
else if $GCD(p) < GCD(q)$ then
    return $p$
else if $GCD(q) < GCD(p)$ then
    return $q$
else if $random(0,1) < 0.5$ then
    return $p$
else
    return $q$
end if
```

GCD). Finally, if *GCD* still fails to distinguish between the two solutions, the tie will be split randomly.

E. Environmental Selection

Environmental selection which aims to obtain a well-approximated and well-distributed archive set is implemented by picking out the "best" solutions from the previous population and the newly created population. A straightforward way to do the selection is based on the fitness of solutions. However, a shortcoming of this way is that it may lead to the loss of diversity since adjacent solutions often have similar fitness values. For example, solutions **E**, **F**, and **G** in Fig. 3 have similar fitness values, and thus a high likelihood of being eliminated or preserved simultaneously. Here, a fitness adjustment mechanism is introduced to address this issue.

1) Fitness Adjustment: GrEA selects individuals by hierarchically comparing them according to the three fitness criteria: *GR*, *GCD*, and *GCPD*. *GR* is the primary criterion, *GCD* is regarded as the secondary one activated when the *GR* value of individuals is incomparable (i.e., equal), and when the first two criteria fail to discriminate individuals, the third one *GCPD* is used to break a tie. Here, we focus the adjustment on the primary criterion.

When an individual is selected into the archive, the *GR* value of its "related" individuals will be punished. However, how to implement the *GR* punishment operation (i.e., determine the "related" individuals and assign how much they would be punished) is not a trivial task. Several crucial factors need to be considered in order to achieve a good balance between convergence and diversity in the archive.

- A severe penalty should be imposed on individuals that have the same grid coordinate as the picked individual.
- The individuals grid-dominated by the picked individual should be punished more heavily than the individuals not grid-dominated by it. For instance, consider a set of individuals **A**, **B**, and **C** which have the grid coordinate $(0,3)$, $(0,5)$, and $(5,0)$, respectively. Obviously, **C** is preferable to **B** after **A** has already entered the archive, because **C** is helpful to the evolution towards different directions.
- In order to further prevent crowding, the neighbors of the picked individual should be penalized, and the punishment degree should decline with the distance from them to the picked individual.
- When implementing penalty on the neighbors of the picked individual, the individuals grid-dominated by them may also need to be punished. For example, for a set of four individuals $\mathbf{A}(0,0,1)$, $\mathbf{B}(0,1,0)$, $\mathbf{C}(1,0,0)$, and $\mathbf{D}(3,0,0)$, assume that three individuals need to be selected into the archive. Apparently, the best choice is to eliminate the last individual **D**. However, individual **C** may fail to be selected after **A** and **B** are in the archive, since punishments were imposed on **C** as the neighbor of **A** and **B**, leading **C** to have a worse *GR* than **D**. Therefore, a punishment on the individuals grid-dominated by the neighbors of the picked individual is advisable, which can improve the convergence of the archive set largely.

Algorithm 3 $GR\_adjustment(P, q)$

```
Require: $P$ (candidate set), $q$ (picked individual), $M$ (number of objectives), $PD(p)$ (maximum punishment degree of $p$), $E(q) := \{p \in P | GD(p,q) = 0\}$, $G(q) := \{p \in P | q \prec_{grid} p\}$, $NG(q) := \{p \in P | q \nprec_{grid} p\}$, $N(q) := \{p \in P | GD(p,q) < M\}$
 1: for all $p \in E(q)$ do
 2:     $GR(p) \leftarrow GR(p) + (M+2)$
 3: end for
 4: for all $p \in G(q)$ do
 5:     $GR(p) \leftarrow GR(p) + M$
 6: end for
 7: for all $p \in NG(q) \wedge p \notin E(q)$ do
 8:     $PD(p) \leftarrow 0$
 9: end for
10: for all $p \in N(q) \cap NG(q) \wedge p \notin E(q)$ do
11:     if $PD(p) < M - GD(p,q)$ then
12:         $PD(p) \leftarrow M - GD(p,q)$
13:         for all $r \in G(p) \wedge r \notin G(q) \cup E(q)$ do
14:             if $PD(r) < PD(p)$ then
15:                 $PD(r) \leftarrow PD(p)$
16:             end if
17:         end for
18:     end if
19: end for
20: for all $p \in NG(q) \wedge p \notin E(q)$ do
21:     $GR(p) \leftarrow GR(p) + PD(p)$
22: end for
```

Keeping the above factors in mind, a *GR* adjustment procedure is presented in Algorithm 3. Clearly, individuals can be classified into three groups in the *GR* adjustment process: the individuals whose grid coordinate is equal to that of the picked individual (lines 1–3), the ones who are grid-dominated by the picked individual (lines 4–6), and the ones who are not grid-dominated by and have a different grid coordinate from the picked individual (lines 7–22). They correspond to the punishment degrees $M+2$, M, and within $[0, M-1]$, respectively, where M denotes the number of objectives. Specifically, for the individuals in the last group, a neighbor p of the picked individual q is imposed the punishment degree at least $M - GD(p,q)$ (lines 11 and 12), and correspondingly the individuals grid-dominated by p is imposed the punishment degree more than or equal to that of p (lines 13–17). This can prevent the individuals from being archived earlier than their better competitors in the sense of the grid dominance relation.

Algorithm 4 $Environmental\_selection(P)$

```
Require: $N$ (archive size)
 1: Generate an empty set $Q$ for archive
 2: $(F_1, F_2, ..., F_i, ...) \leftarrow Pareto\_nondominated\_sort(P)$
          /* Partition $P$ into different fronts $(F_1, F_2, ..., F_i, ...)$ by using the fast nondominated sorting approach and find the critical front $F_i$ (i.e., $0 \leq N - |F_1 \cup F_2 \cup ... \cup F_{i-1}| < F_i$) */
 3: $Q \leftarrow F_1 \cup F_2 \cup ... \cup F_{i-1}$
 4: if $|Q| = N$ then
 5:     return Q
 6: end if
 7: $Grid\_setting(F_i)$          /* Set grid environment for $F_i$ */
 8: $Initialization(F_i)$
 9: while $|Q| < N$ do
10:     $q \leftarrow Findout\_best(F_i)$
11:     $Q \leftarrow Q \cup \{q\}$
12:     $F_i \leftarrow F_i \setminus \{q\}$
13:     $GCD\_calculation(F_i, q)$
14:     $GR\_adjustment(F_i, q)$
15: end while
16: return $Q$
```

To summarize, by the fitness adjustment operation, *GR* will not be viewed as a simple convergence indicator, but rather a combination of information among proximity, density, and evolution direction of individuals in the archive set. Next, we give the main procedure of environmental selection.

2) Main Procedure: Algorithm 4 shows the main procedure of environmental selection. Similar to NSGA-II [11], GrEA considers the critical Pareto nondominated front in the candidate set. The candidate solutions are divided into different fronts $(F_1, F_2, ..., F_i, ...)$ by using the fast nondominated sorting approach. The critical front F_i ($|F_1 \cup F_2 \cup ... \cup F_{i-1}| \leq N$ and $|F_1 \cup F_2 \cup ... \cup F_{i-1} \cup F_i| > N$, where N denotes the archive size) is found, and correspondingly the first $(i-1)$ nondominated fronts $(F_1, F_2, ..., F_{i-1})$ are moved into the archive (lines 2–6). In fact, since the solutions in many-objective problems are Pareto nondominated to each other, the critical front is often the first front, namely $i = 1$.

In Algorithm 4, function $Initialization$ (line 8) is used to initialize the information of individuals in the grid environment set in line 7. The fitness of individuals with regard to convergence (i.e., *GR* and *GCPD*) is calculated by Eqs. (9) and (11). It is necessary to point out that the initial density value of individuals (i.e., *GCD*) is assigned to zero in the function. Unlike the convergence estimator, which can be directly calculated by the own location of an individual, the diversity one has to be estimated by the relation to other individuals. It may be meaningless to consider the crowding relation among the individuals in the candidate set rather than in the archive set, since the latter is only the population to be preserved. Here, GrEA estimates the density of individuals by calculating their crowding degree in the archive (line 13). Algorithms 5 and 6 give the pseudocode of functions $Initialization$ and $GCD\_calculation$, respectively.

Function $Findout\_best$ (line 10) in Algorithm 4 is designed to find out the best individual in the considered front. The pseudocode is shown in Algorithm 7. As stated previously, the function hierarchically compares the three criteria *GR*, *GCD*, and *GCPD*. A lower value is preferable in all criteria.

Algorithm 5 $Initialization(P)$

```
for all $p \in P$ do
    $GR\_assignment(p)$
                /* Assign GR according to equation (9) */
    $GCPD\_assignment(p)$
                /* Assign GCPD according to equation (11) */
    $GCD(p) \leftarrow 0$          /* Assign zero to GCD */
end for
```

Algorithm 6 $GCD\_calculation(P, q)$

```
Require: $P$ (candidate set), $q$ (picked individual), $N(q) := \{p \in P | GD(p, q) < M\}$
for all $p \in N(q)$ do
    $GCD(p) \leftarrow GCD(p) + (M - GD(p, q))$
end for
```

Algorithm 7 $Findout\_best(P)$

```
Require: $q$ (best solution in $P$), $p_i$ (the $i$th solution in $P$)
$q \leftarrow p_1$
for $i = 2$ to $|P|$ do
    if $GR(p_i) < GR(q)$ then
        $q \leftarrow p_i$
    else if $GR(p_i) = GR(q)$ then
        if $GCD(p_i) < GCD(q)$ then
            $q \leftarrow p_i$
        else if $GCD(p_i) = GCD(q)$ then
            if $GCPD(p_i) < GCPD(q)$ then
                $q \leftarrow p_i$
            end if
        end if
    end if
end for
return $q$
```

An example that illustrates the working principle of the whole environmental selection process is given in the supplement document of this paper.

IV. Experimental Design

This section is devoted to the experimental design for investigating the performance of GrEA. We first give the test problems and performance metrics used in the experiment. Then, we briefly introduce five state-of-the-art EMO algorithms: ϵ-MOEA [15], POGA [17], HypE [3], MSOPS [26], and MOEA/D [71] which are used to validate the proposed algorithm. Finally, the general experimental setting is provided for the comparative studies of these algorithms.

A. Test Problems and Performance Metrics

As a basis for the comparisons, the DTLZ [16], DTLZ5(I, M) [60], and multiobjective traveling salesman problem (MOTSP) [9] test problem suites are considered. All these problems can be scaled to any number of objectives and decision variables. In this study, we divided them into four groups according to their characteristics of challenging different abilities of algorithms.

For the DTLZ suite, the problems DTLZ1 to DTLZ7 can be classified into two groups. One group, including DTLZ2,

TABLE I
SETTINGS OF THE TEST PROBLEMS

Name	Number of Objectives (M)	Number of Variables (n)	Parameter
DTLZ1	$4, 5, 6, 8, 10$	$M-1+k$	$k=5$
DTLZ2	$4, 5, 6, 8, 10$	$M-1+k$	$k=10$
DTLZ3	$4, 5, 6, 8, 10$	$M-1+k$	$k=10$
DTLZ4	$4, 5, 6, 8, 10$	$M-1+k$	$k=10$
DTLZ5	$4, 5, 6, 8, 10$	$M-1+k$	$k=10$
DTLZ6	$4, 5, 6, 8, 10$	$M-1+k$	$k=10$
DTLZ7	$4, 5, 6, 8, 10$	$M-1+k$	$k=20$
DTLZ5(I, M)	10	$M-1+k$	$k=10$, $I=3,4,5,6,7,8,9$
MOTSP	$5, 10$	30	$TSPcp=$ $-0.4, -0.2, 0, 0.2, 0.4$

DTLZ4, DTLZ5, and DTLZ7, is used to test the ability of an algorithm to cope with the problems with different shapes and locations. The other group, including DTLZ1, DTLZ3, and DTLZ6, creates more obstacles for an algorithm to converge into the Pareto front [16].

The DTLZ5(I, M) suite, originating from DTLZ5, is a set of test problems where the actual dimensionality I of the Pareto front against the original number M of objectives in the problem can be controlled by the user. In these problems, all objectives within $\{f_1, ..., f_{M-I+1}\}$ are positively correlated, while the objectives in $\{f_{M-I+2}, ..., f_M\}$ are conflicting with each other. A detailed description of the DTLZ and DTLZ5(I, M) problem suites can be found in [16], [60].

The MOTSP is a typical combinatorial optimization problem and can be stated as [9]: given a network $N = (V, C)$, where $V = \{v_1, v_2, ..., v_n\}$ is a set of nodes and $C = \{c_k : k \in \{1, 2, ..., M\}\}$ is a set of cost matrices between nodes ($c_k : V \times V$), determine the Pareto optimal set for the minimum length Hamiltonian cycles. The M matrices, according to [9], can be constructed as follows.

The matrix c_1 is first generated by assigning each distinct pair of nodes with a random number between 0 and 1. Then the matrix c_{k+1} is generated according to the matrix c_k:

$$c_{k+1}(i, j) = TSPcp \times c_k(i, j) + (1 - TSPcp) \times rand \quad (12)$$

where $c_k(i, j)$ denotes the cost from node i to node j in matrix c_k, $rand$ is a function to generate a uniform random number in $[0, 1]$, and $TSPcp \in (-1, 1)$ is a simple TSP "correlation parameter". When $TSPcp < 0$, $TSPcp = 0$, or $TSPcp > 0$, it introduces negative, zero, or positive interobjective correlations, respectively. In our study, $TSPcp$ is assigned to -0.4, -0.2, 0, 0.2, and 0.4 respectively to represent different characteristics of the problem. In each case, the number of nodes (i.e., decision variables) is set to 30.

The summary of settings for all test problems is shown in Table I.

In order to compare the performance of the selected algorithms, we introduce two widely-used quality metrics, inverted generational distance (IGD) [5], [48] and hypervolume (HV) [12]. The former requires a reference set of representing the Pareto front, and is used to evaluate algorithms on DTLZ and DTLZ5(I, M) since their optimal fronts are known. The last one is used to assess the performance of algorithms on MOTSP whose Pareto front is unknown.

A reference point is required in the HV calculation. Here, we regard the point with the integer value 22 for each objective as the reference point, considering that it is slightly larger than the worst value of all the obtained solution sets. In addition, since the exact calculation of the hypervolume metric is infeasible for a solution set with 10 objectives, we approximately estimate the hypervolume result of a solution set by the Monte Carlo sampling method used in HypE [3]. According to [3], 10,000,000 sampling points are used to evaluate the result.

B. Five Other Algorithms in Comparison

In order to validate the proposed GrEA, we consider five state-of-the-art EMO algorithms as the peer algorithms:

- **ϵ-dominance based Multiobjective Evolutionary Algorithm (ϵ-MOEA)** [15]. ϵ-MOEA[1] is a steady-state algorithm using the ϵ-dominance relation. It divides the objective space into hyperboxes by a size of ϵ. Each hyperbox is assigned at most a single point on the basis of ϵ-dominance for different hyperboxes and Pareto dominance for an identical hyperbox. ϵ-MOEA has been found to perform well for many-objective optimization problems in a recent study [21].
- **Preference Order based Genetic Algorithm (POGA)** [17]. POGA[2] uses a preference order-based approach to solve many-objective optimization problems. By considering the concept of *efficiency of order* in the subsets of objectives, POGA provides a higher selection pressure towards the Pareto front than Pareto dominance-based algorithms.
- **Hypervolume Estimation Algorithm (HypE)** [3]. HypE[3] is a Hypervolume-based evolutionary many-objective optimization algorithm. It uses Monte Carlo simulation to approximate the exact hypervolume value, significantly reducing the time cost of the HV calculation and enabling hypervolume-based search to be widely applied on many-objective optimization.
- **Multiple Single Objective Pareto Sampling (MSOPS)** [26]. MSOPS[4] uses the idea of single-objective aggregated optimization to search in parallel for points that lie on the Pareto front. MSOPS is a very popular algorithm to tackle many-objective optimization problems since it can achieve a good balance between convergence and diversity [27], [66].
- **Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D)** [71]. MOEA/D[5], one of the most popular EMO algorithms developed recently, is also an aggregation-based algorithm. Unlike MSOPS where a number of weight vectors correspond to an individual, MOEA/D specifies an individual with only one weight vector. MOEA/D has demonstrated its ad-

[1] The code of ϵ-MOEA is available at *http://www.iitk.ac.in/kangal*.

[2] The code of POGA is written by ourselves.

[3] The code of HypE is available at *http://www.tik.ee.ethz.ch/pisa*.

[4] The code of MSOPS is available at *http://code.evanhughes.org/*.

[5] The code of MOEA/D is available at *http://dces.essex.ac.uk/staff/zhang /webofmoead.htm*.

TABLE II
PARAMETER SETTINGS IN GREA AND ϵ-MOEA, WHERE M IS THE SUMBER OF OBJECTIVES

Problem	$M = 4$		$M = 5$		$M = 6$		$M = 8$		$M = 10$		Problem	$M = 10$		Problem	$M = 5$		$M = 10$	
	div	ϵ	div	ϵ	div	ϵ	div	ϵ	div	ϵ	(*I*, *M*)	div	ϵ	(*TSPcp*)	div	ϵ	div	ϵ
DTLZ1	10	0.0520	10	0.0590	10	0.0554	10	0.0549	11	0.0565	DTLZ5(3,10)	9	0.06	MOTSP(–0.4)	12	2.4	11	6.5
DTLZ2	10	0.1312	9	0.1927	8	0.2340	7	0.2900	8	0.3080	DTLZ5(4,10)	10	0.12	MOTSP(–0.2)	11	1.9	10	4.8
DTLZ3	11	0.1385	11	0.2000	11	0.2270	10	0.1567	11	0.8500	DTLZ5(5,10)	10	0.16	MOTSP(0)	11	1.4	10	3.8
DTLZ4	10	0.1312	9	0.1927	8	0.2340	7	0.2900	8	0.3080	DTLZ5(6,10)	10	0.2	MOTSP(0.2)	11	0.95	10	2.6
DTLZ5	35	0.0420	29	0.0785	14	0.1100	11	0.1272	11	0.1288	DTLZ5(7,10)	8	0.24	MOTSP(0.4)	10	0.6	10	1.8
DTLZ6	36	0.1200	24	0.3552	50	0.7500	50	1.1500	50	1.4500	DTLZ5(8,10)	8	0.25					
DTLZ7	9	0.1050	8	0.1580	6	0.1500	5	0.2250	4	0.5600	DTLZ5(9,10)	9	0.26					

vantage in both multi-objective [47] and many-objective optimization [30].

C. General Experimental Setting

Parameter settings for all conducted experiments are given as follows unless otherwise mentioned.

- **Parameter setting for crossover and mutation**: A crossover probability $p_c = 1.0$ and a mutation probability $p_m = 1/n$ (where n denotes the number of decision variables) were used. For DTLZ and DTLZ5(I, M), the operators for crossover and mutation are simulated binary crossover (SBX) and polynomial mutation with both distribution indexes 20 (i.e., $\eta_c = 20$ and $\eta_m = 20$) [12]. As to MOTSP, the order crossover (OX) and inversion operator [52] are chosen as crossover and mutation operators, respectively.
- **Number of runs and stopping condition**: We independently run each algorithm 30 times on each test problem. The termination criterion of an algorithm is a predefined number of evaluations. For the first problem group (DTLZ2, DTLZ4, DTLZ5, and DTLZ7) and the third problem group (DTLZ5(I, M)), it was set to 30,000, and for the second group (DTLZ1, DTLZ3, and DTLZ6) and the fourth group (MOTSP), it was set to 100,000.
- **Population and archive size**: For general EMO algorithms, the population size was set to 100, and the archive was also maintained with the same size if required. Note that the population size, in MOEA/D, is the same as the number of weight vectors. Due to the combinatorial nature of uniformly distributed weight vectors, the population size cannot be arbitrarily specified. Here, we use the closest integer to 100 among the possible values as the population size (i.e., 120, 126, 126, 120, and 55 for 4-, 5-, 6-, 8-, and 10-objective problems, respectively). In ϵ-MOEA, the population size is determined by the ϵ value. In order to guarantee a fair comparison, we set ϵ so that the archive of ϵ-MOEA is approximately of the same size as that of the other algorithms (shown in Table II).
- **Parameter setting in MSOPS, MOEA/D, and GrEA**: In MSOPS, the number of weight vectors was set to 100 as suggested in [66]. Following the practice in [33], the Tchebycheff function in MOEA/D was selected as the scalarizing function and the neighborhood size was specified as 10% of the population size. In GrEA, the setting of grid division div is shown in Table II. A detailed study of div will be given in Section V-B.

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we validate the performance of GrEA according to the experimental design described in the previous section. Our experiments can be divided into two parts. The first one is to compare GrEA with the five state-of-the-art EMO algorithms. The second one is to investigate the effect of the grid division parameter in the proposed algorithm.

A. Performance Comparison

1) The DTLZ2, DTLZ4, DTLZ5, and DTLZ7 Test Problems: Table III shows the IGD results in terms of the mean and standard deviation over 30 runs of the six EMO algorithms on DTLZ2, DTLZ4, DTLZ5, and DTLZ7, where the best mean for each problem is shown with a gray background. As can be seen from the table, GrEA performs the best on DTLZ2 and DTLZ7 with respect to all considered numbers of objectives. For the other problems, MSOPS and GrEA have their own strong points. For DTLZ4, MSOPS obtains the best IGD value in the low-dimension objective space, and GrEA outperforms the other algorithms when the number of objectives is larger than four. For DTLZ5, GrEA performs well on the 4-objective problem, and MSOPS obtains better results with the increase of the number of objectives.

Note that the MOEA/D algorithm, which has recently been found to be very successful in the EMO community, obtains a worse IGD value than GrEA, MSOPS, and ϵ-MOEA in general. In fact, the solution set obtained by MOEA/D is very close to the Pareto front for some test problems, such as DTLZ2 and DTLZ4 (the convergence and diversity comparison results of the six algorithms can be found in the supplement document of the paper). But it has a worse coverage of the Pareto front than that obtained by GrEA, MSOPS, and ϵ-MOEA for most of the problems, thereby resulting in a worse IGD value on these problems. This occurrence may be attributed to the aggregation-based selection operation in MOEA/D. Although a set of uniformly-distributed weight vectors is selected to specify the search targets (i.e., the points on the Pareto front), it cannot ensure that these points are located uniformly, especially for some problems with irregular shape. In addition, the Tchebycheff-based scalarizing function may not be a good tool to maintain diversity of solutions. Many weight vectors may correspond to only one Pareto-optimal point by this scalarizing function [13]. Similar observations have been reported in [13], [33].

Further studies with these algorithms have been conducted to exhibit their evolutionary trajectories. Figure 4 plots the

TABLE III
IGD RESULTS OF THE SIX ALGORITHMS ON DTLZ2, DTLZ4, DTLZ5, AND DTLZ7

Problem	Obj.	ϵ-MOEA	POGA	HypE	MSOPS	MOEA/D	GrEA
DTLZ2	4	1.348E–1 (2.4E–3)	1.486E–1 (6.7E–3)	2.452E–1 (3.9E–2)	1.757E–1 (1.2E–2)	2.124E–1 (2.1E–3)	1.271E–1 (2.5E–3)
	5	1.945E–1 (1.2E–2)	2.758E–1 (1.8E–2)	4.245E–1 (7.5E–2)	3.625E–1 (2.8E–2)	2.674E–1 (1.1E–3)	1.750E–1 (2.9E–3)
	6	3.161E–1 (6.7E–3)	5.882E–1 (5.8E–2)	4.738E–1 (5.7E–2)	3.698E–1 (2.2E–2)	4.132E–1 (1.9E–2)	2.985E–1 (5.2E–3)
	8	4.408E–1 (1.3E–2)	9.903E–1 (9.3E–2)	6.147E–1 (5.0E–2)	5.681E–1 (3.6E–2)	6.365E–1 (7.1E–2)	3.957E–1 (4.6E–3)
	10	5.323E–1 (2.4E–2)	1.162E+0 (1.0E–1)	6.972E–1 (5.5E–2)	7.658E–1 (4.3E–2)	7.257E–1 (6.5E–2)	4.842E–1 (2.9E–3)
DTLZ4	4	4.150E–1 (2.7E–1)	2.663E–1 (6.8E–2)	4.976E–1 (3.4E–1)	1.449E–1 (4.4E–3)	5.382E–1 (2.8E–1)	1.913E–1 (1.1E–1)
	5	6.330E–1 (3.4E–1)	3.970E–1 (7.0E–2)	7.018E–1 (2.8E–1)	3.147E–1 (3.2E–2)	5.857E–1 (3.0E–1)	2.154E–1 (9.7E–2)
	6	6.035E–1 (1.7E–1)	9.541E–1 (1.6E–1)	6.672E–1 (1.0E–1)	3.685E–1 (1.1E–2)	6.498E–1 (1.6E–1)	3.007E–1 (4.6E–3)
	8	6.459E–1 (1.1E–1)	1.175E+1 (2.2E–1)	9.199E–1 (6.1E–2)	5.400E–1 (2.5E–2)	7.600E-1 (8.6E–2)	4.020E–1 (3.2E–3)
	10	6.267E–1 (9.2E–2)	8.668E–1 (2.7E–1)	1.074E+0 (6.1E–2)	8.178E–1 (4.5E–2)	8.311E–1 (8.4E–2)	4.928E–1 (3.9E–3)
DTLZ5	4	4.819E–2 (5.3E–3)	5.217E–2 (9.8E–3)	1.216E–1 (4.1E–2)	3.016E–2 (3.0E–3)	2.563E–2 (1.1E–4)	1.846E–2 (3.3E–3)
	5	8.956E–2 (7.5E–3)	7.437E–1 (1.6E–2)	1.459E–1 (5.3E–2)	3.002E–2 (4.1E–3)	4.544E–2 (1.5E–3)	4.333E–2 (2.2E–2)
	6	1.278E–1 (1.1E–2)	7.467E–1 (9.8E–4)	1.734E–1 (5.9E–2)	1.859E–2 (1.7E–3)	6.947E–2 (4.3E–3)	9.575E–2 (1.6E–2)
	8	1.589E–1 (2.1E–2)	7.473E–1 (1.3E–3)	1.754E–1 (6.9E–1)	2.525E–2 (2.5E–3)	1.088E–1 (7.4E–3)	2.327E–1 (3.5E–2)
	10	1.690E–1 (2.1E–2)	1.219E+0 (7.7E–1)	1.560E–1 (4.8E–2)	4.025E–2 (3.9E–3)	1.959E–1 (1.3E–2)	3.462E–1 (6.1E–2)
DTLZ7	4	3.492E–1 (1.8E–1)	2.151E–1 (8.6E–3)	4.846E–1 (1.9E–1)	1.547E+0 (4.1E–1)	5.152E–1 (7.2E–2)	1.897E–1 (6.8E–3)
	5	6.310E–1 (2.1E–1)	4.127E–1 (1.6E–2)	8.972E–1 (2.1E–1)	7.581E+0 (1.3E+0)	6.444E–1 (8.7E–2)	3.238E–1 (1.0E–2)
	6	5.856E–1 (1.9E–1)	6.608E–1 (3.0E–2)	9.894E–1 (1.9E–1)	1.085E+1 (2.5E+0)	7.551E–1 (6.1E–2)	4.888E–1 (1.6E–2)
	8	8.971E–1 (5.1E–1)	2.293E+0 (4.1E–1)	1.065E+0 (4.1E–2)	1.945E+1 (2.0E+0)	1.064E+0 (1.3E–1)	7.643E–1 (3.5E–2)
	10	1.180E+0 (3.6E–1)	4.211E+0 (7.5E–1)	1.224E+0 (8.1E–2)	2.670E+1 (3.4E+0)	1.546E+0 (2.0E–1)	1.057E+0 (3.8E–2)

TABLE IV
IGD RESULTS OF THE SIX ALGORITHMS ON DTLZ1, DTLZ3, AND DTLZ6

Problem	Obj.	ϵ-MOEA	POGA	HypE	MSOPS	MOEA/D	GrEA
DTLZ1	4	4.866E–2 (2.3E–3)	9.301E–1 (3.7E–2)	1.355E–1 (7.1E–2)	5.762E–2 (3.0E–3)	9.653E–2 (1.1E–4)	4.624E–2 (5.3E–3)
	5	6.930E–2 (8.1E–3)	4.470E+0 (4.3E+0)	3.107E–1 (3.8E–1)	8.673E–2 (3.8E–3)	1.192E–1 (1.3E–4)	6.257E–2 (7.5E–3)
	6	9.877E–2 (8.4E–2)	1.284E+1 (1.7E+1)	5.554E–1 (5.9E–1)	1.808E–1 (1.1E–1)	1.377E–1 (6.4E–3)	8.543E–2 (1.0E–2)
	8	3.069E–1 (3.4E–1)	1.012E+1 (6.9E+0)	1.017E+0 (1.4E+0)	7.770E–1 (6.8E-1)	1.852E–1 (5.8E–3)	1.060E–1 (4.8E–3)
	10	4.071E–1 (3.8E–1)	1.370E+1 (9.2E+0)	1.485E+0 (1.8E+0)	1.623E+0 (1.0E+0)	2.188E–1 (6.4E–3)	2.864E–1 (1.0E–1)
DTLZ3	4	1.417E–1 (8.9E–3)	1.317E+1 (5.7E+0)	1.029E+0 (7.1E–1)	1.072E+1 (6.8E+0)	2.130E–1 (1.7E–3)	1.522E–1 (4.7E–2)
	5	2.267E–1 (3.3E–2)	2.341E+1 (1.1E+1)	4.708E+0 (5.6E+0)	2.896E+1 (1.5E+1)	2.677E–1 (7.8E–4)	2.804E–1 (8.3E–2)
	6	4.578E–1 (1.4E–1)	3.239E+1 (1.2E+1)	2.689E+0 (1.7E+0)	4.665E+1 (1.6E+1)	4.085E–1 (3.2E–2)	4.368E–1 (1.5E–1)
	8	1.122E+1 (1.4E+1)	2.733E+1 (1.2E+1)	7.476E+0 (9.6E+0)	6.095E+1 (1.9E+1)	6.106E–1 (7.5E–2)	5.546E–1 (2.3E–1)
	10	2.040E+1 (2.7E+1)	3.157E+1 (1.4E+1)	6.226E+0 (6.2E+0)	6.312E+1 (1.8E+1)	6.599E–1 (5.5E–2)	7.743E–1 (2.5E–1)
DTLZ6	4	4.671E–1 (2.8E–2)	2.226E+0 (3.0E–1)	3.919E+0 (6.5E–1)	4.178E+0 (6.4E–1)	8.015E–2 (2.7E–2)	7.045E–2 (3.1E–2)
	5	1.678E+0 (1.5E–1)	1.801E+0 (4.5E–1)	5.431E+0 (5.6E–1)	6.560E+0 (5.0E–1)	1.194E–1 (3.8E–2)	1.429E–1 (4.3E–2)
	6	2.705E+0 (2.8E–1)	2.245E+0 (6.0E–1)	5.679E+0 (5.7E–1)	6.862E+0 (5.1E–1)	1.569E–1 (3.7E–2)	4.544E–1 (9.1E–2)
	8	1.987E+0 (1.3E+0)	5.807E+0 (3.9E+0)	6.165E+0 (6.2E–1)	6.813E+0 (4.4E–1)	1.830E–1 (2.8E–2)	5.971E–1 (3.8E–1)
	10	3.737E+0 (2.0E+0)	8.941E+0 (1.4E–1)	6.428E+0 (4.1E–1)	6.728E+0 (4.8E–1)	2.692E–1 (3.0E–2)	9.432E–1 (7.8E–1)

performance trajectories of IGD for the six algorithms on the 10-objective DTLZ2. Clearly, GrEA performs better than the other five algorithms. Although ϵ-MOEA outperforms GrEA in the initial stage of evolution, the latter exceeds the former at around 6,000 evaluations and keeps a clear advantage until the end.

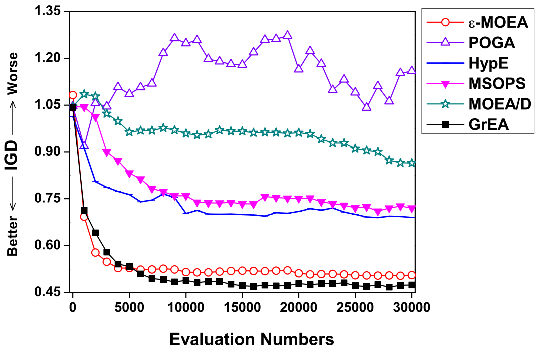

Fig. 4. Evolutionary trajectories of IGD for the six algorithms on the ten-objective DTLZ2.

2) The DTLZ1, DTLZ3, and DTLZ6 Test Problems: The IGD results of the six EMO algorithms on this problem group are shown in Table IV. It is clear that the algorithms GrEA and MOEA/D in general outperform the other four algorithms. They obtain the best IGD value in 6 and 7 out of 15 test instances, respectively. More specifically, GrEA performs the best on the 8-objective DTLZ3, 4-objective DTLZ6, and DTLZ1 for all considered numbers of objectives except 10. MOEA/D outperforms the other algorithms on the 10-objective DTLZ1, 6- and 10-objective DTLZ3, and DTLZ6 with the number of objectives larger than 4. For the rest of the problems (i.e., the 4- and 5-objective DTLZ3), ϵ-MOEA reaches the best result.

It is interesting to note that MOEA/D, unlike on the first group of test problems, is very competitive on this group of

TABLE V
IGD RESULTS OF THE SIX ALGORITHMS ON DTLZ5(I, M), WHERE $M = 10$.

I	ϵ-MOEA	POGA	HypE	MSOPS	MOEA/D	GrEA
3	6.435E–2 (2.2E–3)	1.810E+1 (1.9E+1)	2.068E–1 (2.5E–2)	5.433E–1 (5.1E–2)	3.060E–1 (8.7E–3)	1.418E–1 (6.7E–2)
4	1.402E–1 (1.7E–2)	2.271E+1 (1.7E+1)	2.997E–1 (2.9E–2)	3.291E–1 (9.7E–4)	3.586E–1 (3.3E–2)	1.528E–1 (8.8E–3)
5	2.605E–1 (1.1E–1)	1.354E+1 (1.7E+1)	3.923E–1 (9.4E–2)	2.982E–1 (9.4E–3)	3.753E–1 (5.0E–2)	1.932E–1 (1.4E–2)
6	4.024E–1 (1.2E–1)	8.689E+0 (9.6E+0)	6.885E–1 (1.2E–1)	6.031E–1 (1.1E–1)	5.838E–1 (7.5E–2)	3.214E–1 (1.5E–2)
7	4.894E–1 (1.3E–1)	1.061E+1 (1.1E+1)	7.911E–1 (1.0E–1)	7.934E–1 (1.1E–1)	6.343E–1 (8.6E–2)	3.808E–1 (1.1E–2)
8	5.119E–1 (9.0E–2)	8.520E+1 (9.8E+0)	8.584E–1 (8.9E–2)	9.273E–1 (1.1E–1)	6.520E–1 (6.5E–2)	4.352E–1 (6.8E–2)
9	5.694E–1 (1.1E–2)	5.823E+1 (9.0E+0)	9.180E–1 (7.5E–2)	1.062E+0 (1.3E–1)	7.411E–1 (7.6E–2)	4.564E–1 (5.1E–3)

TABLE VI
HV RESULTS OF THE SIX ALGORITHMS ON THE MOTSP.

Obj.	$TSPcp$	ϵ-MOEA	POGA	HypE	MSOPS	MOEA/D	GrEA
5	–0.4	1.125E+6 (5.3E+4)	1.128E+6 (5.2E+4)	1.962E+5 (5.3E+4)	7.963E+5 (4.5E+4)	9.615E+5 (5.8E+4)	1.231E+6 (4.8E+4)
	–0.2	1.014E+6 (4.5E+4)	1.012E+6 (3.7E+4)	2.493E+5 (4.8E+4)	8.048E+5 (4.2E+4)	9.687E+5 (4.8E+4)	1.121E+6 (3.8E+4)
	0	9.211E+5 (2.7E+4)	8.445E+5 (3.2E+4)	3.596E+5 (4.6E+4)	7.837E+5 (3.0E+4)	8.729E+5 (2.8E+4)	9.844E+5 (3.5E+4)
	2	8.543E+5 (3.1E+4)	7.565E+5 (3.2E+4)	4.251E+5 (6.7E+4)	7.651E+5 (3.4E+4)	7.530E+5 (3.1E+4)	8.836E+5 (2.3E+4)
	4	8.177E+5 (2.1E+4)	7.206E+5 (4.8E+4)	5.259E+5 (5.5E+4)	7.737E+5 (2.5E+4)	7.255E+5 (4.3E+4)	8.466E+5 (2.4E+4)
10	–0.4	1.026E+11 (1.4E+10)	2.520E+11 (5.7E+10)	3.634E+09 (1.7E+09)	1.706E+11 (2.4E+10)	1.539E+10 (3.2E+09)	3.772E+11 (1.6E+10)
	–0.2	1.203E+11 (1.3E+10)	1.761E+11 (4.9E+10)	1.198E+10 (3.7E+09)	1.820E+11 (1.8E+10)	3.376E+10 (5.1E+09)	3.097E+11 (1.0E+10)
	0	1.260E+11 (1.5E+10)	1.181E+11 (4.1E+10)	2.386E+10 (7.0E+09)	1.704E+11 (1.2E+10)	5.458E+10 (8.8E+09)	2.551E+11 (8.3E+09)
	2	1.400E+11 (1.2E+10)	8.500E+10 (2.5E+10)	3.404E+10 (8.3E+09)	1.529E+11 (1.0E+10)	7.450E+10 (1.1E+10)	2.136E+11 (5.9E+09)
	4	1.467E+11 (1.1E+10)	8.095E+10 (1.7E+10)	5.298E+10 (9.2E+09)	1.447E+11 (6.8E+09)	1.068E+11 (1.3E+10)	1.818E+11 (6.8E+09)

test problems which provide more obstacles for an algorithm to converge into the Pareto front. An important reason is that in contrast to some of the other algorithms (such as POGA, HypE, and MSOPS) whose solution set fails to approximate the Pareto front, MOEA/D still works well in terms of convergence for most of the problems in this group, thereby obtaining better IGD results.

3) The DTLZ5(I, M) Test Problem: In this section, we concentrate on the DTLZ5(I, M) problem suite, which tests the ability of an EMO algorithm to find a lower-dimensional Pareto front while working with a higher-dimensional objective space. Table V shows the IGD results of the six algorithms on DTLZ5$(I, 10)$, where I ranges from 3 to 9.

Clearly, GrEA and ϵ-MOEA perform better than the other four algorithms. Specifically, ϵ-MOEA outperforms the other algorithms when I is equal to 3 or 4, and GrEA performs the best for the rest of the problems. MOEA/D, which obtains the best convergence results for all test instances of this problem group (see the supplement document of this paper), has a poor coverage of the Pareto front, leading to a worse IGD result than GrEA and ϵ-MOEA.

4) The MOTSP Test Problem: One important property of the MOTSP problem is that the conflict degree among the objectives can be adjusted according to the parameter $TSPcp \in (-1, 1)$, where a lower value means a greater degree of conflict. From the HV results shown in Table VI, it can be seen that GrEA significantly outperforms the other algorithms for all cases.

In addition, for this combinatorial optimization problem the compared algorithms show some different behavior from for the continuous ones. POGA, which generally performs the worst in the previous test problems, works well when a greater conflict degree among the objectives of MOTSP is involved. ϵ-MOEA performs well for both the 5- and 10-objective problems, and HypE always obtains the worst HV value for all 10 instances. MSOPS, which does not work very well in the 5-objective MOTSP, takes the second place for most of the problems with 10 objectives. The result of MOEA/D for the 10-objective MOTSP is not as good as that for the 5-objective problem, with a worse HV value than MSOPS and ϵ-MOEA.

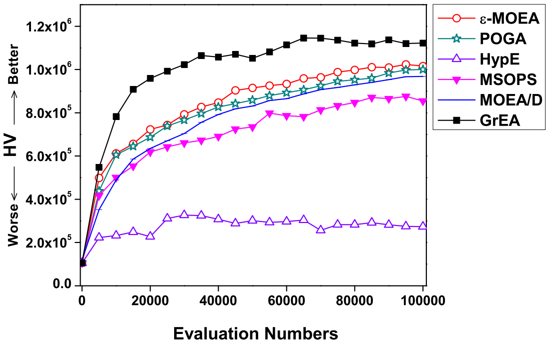

Fig. 5. Evolutionary trajectories of HV for the six algorithms on the five-objective MOTSP, where $TSPcp = -0.2$.

In order to demonstrate the evolutionary process of the six algorithms, Fig. 5 plots their trajectories of HV during 100,000 evaluations on the 5-objective MOTSP with $TSPcp = -0.2$. As can be seen from the figure, the HV trajectory of GrEA rapidly increases in the initial stage of evolution, and keeps a clear advantage over the other algorithms during the whole evolutionary process.

Overall, from the study on the problems with different characteristics, we can conclude that the proposed algorithm has been successful in providing a balance between convergence and diversity in many-objective optimization. GrEA

TABLE VII
PERFORMANCE OF GREA WITH DIFFERENT NUMBER OF DIVISIONS ON THE SIX-OBJECTIVE DTLZ2

Division	5	7	8	9	10	12	15	20	30	50
GD	3.944E–3 (1.0E–3)	3.643E–3 (7.5E–4)	3.475E–3 (8.0E–4)	2.724E–3 (7.0E–4)	2.112E–3 (6.6E–4)	1.823E–3 (6.1E–4)	1.729E–3 (6.1E–4)	1.516E–3 (4.5E–4)	1.464E–3 (4.6E–4)	1.110E–3 (4.4E–4)
DM	7.145E–1 (3.7E–2)	8.812E–1 (2.8E–2)	8.834E–1 (3.2E–2)	8.363E–1 (2.5E–2)	7.713E–1 (2.8E–2)	6.886E–1 (3.6E–2)	6.436E–1 (4.2E–2)	4.654E–1 (4.4E–2)	3.542E–1 (7.6E–1)	1.852E–1 (1.4E–1)
IGD	3.509E–1 (8.1E–3)	2.993E–1 (4.8E–3)	2.985E–1 (5.2E–3)	3.172E–1 (4.4E–3)	3.357E–1 (5.7E–3)	3.647E–1 (3.4E–3)	4.115E–1 (3.5E–3)	4.576E–1 (5.9E–3)	5.820E–1 (2.0E–2)	7.155E–1 (1.3E–1)

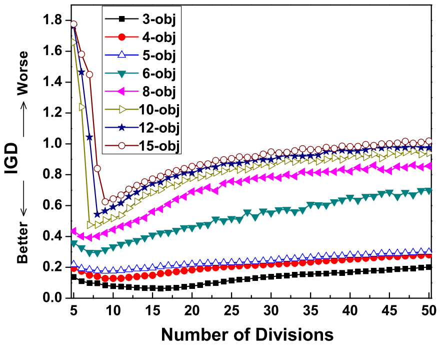

Fig. 6. IGD of GrEA with different number of divisions on DTLZ2.

outperforms the other five state-of-the-art algorithms in 36 out of all 52 test instances.

Due to space limitation, the investigation of the computational complexity of GrEA and the comparison between GrEA and its predecessor are given in the supplement document of the paper.

B. Study of Different Parameter Configurations

In GrEA, a parameter, the grid division (div), is introduced to divide the grid environment. This section investigates the effect of div and try to provide a proper setting for the user. Here, we show the results for the DTLZ2 problem. Similar results can be obtained for other test cases.

To study the sensitivity of the proposed algorithm to div, we repeat the experiments carried out in the previous section for $div \in [5, 50]$ with a step size 1. All other control parameters are kept unchanged. In addition, we expand the number of objectives for the problem to make a clearer observation of GrEA's performance with the variation of div. Figure 6 shows the results of IGD for different divisions on the problem with 3, 4, 5, 6, 8, 10, 12, and 15 objectives.

It is clear from the figure that the IGD value, in general, varies regularly with the number of divisions. In most cases, the trajectory of performance ranging from divisions 5 to around 9 rapidly decreases and then gradually rises until the boundary. Moreover, the sensitivity of the algorithm increases with the number of objectives. For the number of objectives under eight, the performance trajectory remains smooth, and the proposed algorithm appears to perform well during a segment of the range of divisions. When the number of objectives is larger than eight, the effect of division becomes more obvious. A slight variation of div may result in a huge change in the performance of the algorithm. This indicates that a more careful setting of divisions should be conducted for a problem with a larger number of objectives.

On the other hand, considering the most suitable divisions for different numbers of objectives, the best values are distinguishing but similar in general. GrEA performs the best at 16, 9, 9, 8, 7, 7, 8, and 9 divisions for the 3-, 4-, 5-, 6-, 8-, 10-, 12-, and 15-objective DTLZ2 problems, respectively. This phenomenon may be due to the adaptability of the fitness adjustment strategy in the algorithm that tunes the *GR* of individuals adaptively according to the number of objectives.

IGD is a comprehensive performance metric that covers both goals (minimizing the distance to the Pareto front and maximizing the distribution over the optimal front) of an EMO algorithm, but fails to reflect them separately. In the following, we further study the effect of the grid division parameter via separately investigating its effect on convergence and diversity. In our study, two widely-used quality metrics, convergence measure (CM) and diversity measure (DM) [14], are selected. CM assesses the convergence of a solution set by calculating the average normalized Euclidean distance from the set to the Pareto front. A low value is preferable. DM measures the diversity of a solution set by comparing it with a reference set representing the Pareto front. It takes the value between zero and one, and a larger value means a better coverage of the Pareto front. Detailed descriptions of these two metrics can be found in [14]. In addition, in some trials, we observed that the problems with different objectives have similar results with the variation of divisions. Here, only the results on the 6-objective DTLZ2 are demonstrated for brevity.

Table VII shows the results of CM and DM for different division settings on the 6-objective DTLZ2. Clearly, the number of divisions has different influence on convergence and diversity. For CM, the algorithm performs better with the increase of the number of divisions, although the level of variation is not remarkable—it performs well even if the number of divisions falls to five. The main influence is the distribution results of the algorithm. GrEA can maintain "good" diversity during the middle segment of the range of divisions, but when the number of divisions is smaller than 7 or greater than 10, poor performance will be obtained.

This occurrence is probably due to the fitness assignment and adjustment strategies in GrEA. A large hyperbox size (i.e., a small division) would make many solutions located in a single hyperbox and hence assigned the same *GR* and *GCD* values (i.e., the former two criteria of fitness). In this case, the third criterion *GCPD* is activated to distinguish solutions and further guide them evolving and gathering around the utopia point of a unit hyperbox. On the other hand, a small hyperbox size would lead to the increase of the difference of solutions in terms of the criterion *GR*. This means that the degree of the punishment for preventing crowding in the environmental selection process is decreased relatively, thus resulting in the failure of solutions covering the Pareto front.

Overall, although the performance of GrEA varies with the number of grid divisions, it can achieve a good balance between convergence and diversity under a proper setting. Our experiments suggest that a division value around 9 may be reliable on an unknown optimization problem. Furthermore, a slightly larger *div* is recommended if the problem in hand is found to be hard to converge, and a slightly lower value may be more suitable if the coverage of the solutions to the Pareto front is more emphasized.

VI. Conclusion

This paper exploits the potential of grid to handle many-objective optimization problems. The proposed algorithm, GrEA, can mainly be characterized as: 1) executing individual-centered calculation instead of grid-centered calculation throughout the algorithm; 2) increasing the selection pressure towards the optimal front by introducing three grid-based relations: *GR*, *GCPD*, and *grid dominance*; 3) estimating the density of individuals by using adaptive neighborhood whose range varies with the number of objectives; and 4) adjusting the fitness of individuals in the environmental selection process considering both neighborhood and grid dominance relations.

Systematic experiments have been carried out to make an extensive comparison of GrEA with five state-of-the-art EMO algorithms. Several groups of widely used test problems are chosen for challenging different abilities of the algorithms. The results reveal that GrEA is very competitive with the other algorithms in terms of finding a well-approximated and well-distributed solution set in many-objective optimization. Furthermore, the effect of a key parameter, the number of grid divisions, on GrEA has been experimentally investigated. The results show that although the performance of GrEA varies with the number of grid divisions, GrEA can achieve a good tradeoff among convergence and diversity under a proper setting. The division 9 is recommended for an unknown optimization problem, and a slightly higher (or lower) value is suggested when the problem is found to be hard to achieve a good approximation (or coverage) of the Pareto front.

One major future work is to further investigate the proposed algorithms in more MOPs with different characteristics from our previous study, such as WFG [25] and some real-world problems [13], [63]. Moreover, a deeper insight into the behavior of the algorithm is also the focus of our future study. In this context, the effects of the population size and the punishment degree in environmental selection will be first investigated.

Acknowledgement

The authors would like to thank Prof. Q. Zhang for his valued suggestions. They are also grateful to Mr. R. Shen for the help in the experimental study of this work.

References

[1] S. F. Adra and P. J. Fleming, “Diversity management in evolutionary many-objective optimization,” *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 183–195, 2011.

[2] H. Aguirre and K. Tanaka, “Space partitioning with adaptive ϵ-ranking and substitute distance assignments: A comparative study on many-objective MNK-landscapes,” in *Proc. 11th Annual Conf. Genetic Evol. Comput., GECCO 09*, 2009, pp. 547–554.

[3] J. Bader and E. Zitzler, “HypE: An algorithm for fast hypervolume-based many-objective optimization,” *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Jul. 2011.

[4] N. Beume, B. Naujoks, and M. Emmerich, “SMS-EMOA: Multiobjective selection based on dominated hypervolume,” *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[5] P. A. N. Bosman and D. Thierens, “The balance between proximity and diversity in multiobjective evolutionary algorithms,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, 2003.

[6] J. Branke, T. Kaußler, and H. Schmeck, “Guidance in evolutionary multi-objective optimization,” *Advances in Engineering Software*, vol. 32, no. 6, pp. 499–507, 2001.

[7] K. Bringmann and T. Friedrich, “An efficient algorithm for computing hypervolume contributions,” *Evol. Comput.*, vol. 18, pp. 383–402, September 2010.

[8] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler, “On the effects of adding objectives to plateau functions,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 591–603, 2009.

[9] D. W. Corne and J. D. Knowles, “Techniques for highly multiobjective optimisation: some nondominated points are better than others,” in *Proc. 9th Annual Conf. Genetic Evol. Comput.,*, 2007, pp. 773–780.

[10] D. W. Corne, J. D. Knowles, and M. J. Oates, “The Pareto envelope-based selection algorithm for multiobjective optimization,” in *Parallel Problem Solving from Nature PPSN VI*, 2000, pp. 839–848.

[11] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, 2002.

[12] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: John Wiley, 2001.

[13] K. Deb and H. Jain, “An improved NSGA-II procedure for many-objective optimization part I: solving problems with box constraints,” KanGAL, Indian Institute of Technology, Tech. Rep. 2012009, 2012.

[14] K. Deb and S. Jain, “Running performance metrics for evolutionary multi-objective optimization,” KanGAL, Indian Institute of Technology, Tech. Rep. 2002004, 2002.

[15] K. Deb, M. Mohan, and S. Mishra, “Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions,” *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, Dec. 2005.

[16] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable test problems for evolutionary multiobjective optimization,” in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, A. Abraham, L. Jain, and R. Goldberg, Eds., 2005, pp. 105–145.

[17] F. di Pierro, S.-T. Khu, and D. A. Savić, “An investigation on preference order ranking scheme for multiobjective evolutionary optimization,” *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 17–45, 2007.

[18] M. Farina and P. Amato, “A fuzzy definition of ‘optimality’ for many-criteria optimization problems,” *IEEE Tran. Syst. Man, Cybern., Part A: Syst. Hum.*, vol. 34, no. 3, pp. 315–326, 2004.

[19] P. Fleming, R. Purshouse, and R. Lygoe, “Many-objective optimization: an engineering design perspective,” in *Evolutionary Multi-Criterion Optimization*. Springer Berlin / Heidelberg, 2005, pp. 14–32.

[20] M. Garza-Fabre, G. Toscano-Pulido, and C. A. C. Coello, “Two novel approaches for many-objective optimization,” in *Proc. Congr. Evol. Comput.*, 2010, pp. 1–8.

[21] D. Hadka and P. Reed, “Diagnostic assessment of search controls and failure modes in many-objective evolutionary optimization,” *Evol. Comput.*, 2012, in press.

[22] A. G. Hernández-Díaz, L. V. Santana-Quintero, C. A. Coello Coello, and J. Molina, “Pareto-adaptive ϵ-dominance,” *Evol. Comput.*, vol. 15, pp. 493–517, December 2007.

[23] J. G. Herrero, A. Berlanga, and J. M. M. López, “Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 151–168, 2009.

[24] V. L. Huang, A. K. Qin, K. Deb, E. Zitzler, P. N. Suganthan, J. J. Liang, M. Preuss, and H. S., “Problem definitions for performance assessment

of multi-objective optimization algorithms," Nanyang Technological University, Singapore, Tech. Rep., 2007.

[25] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, 2006.

[26] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. Congr. Evol. Comput.*, vol. 4, 2003, pp. 2678–2684.

[27] ——, "Evolutionary many-objective optimisation: many once or one many?" in *Proc. IEEE Congr. Evol. Comput.*, vol. 1, 2005, pp. 222–227.

[28] ——, "Many-objective directed evolutionary line search," in *Proc. 13th Annual Conf. Genetic Evol. Comput.*, 2011, pp. 761–768.

[29] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?" in *Proc. Congr. Evol. Comput.*, vol. 2, 2001, pp. 957–962.

[30] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, 2009, pp. 1758–1763.

[31] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, 2008, pp. 2419–2426.

[32] H. Ishibuchi, N. Akedo, and Y. Nojima, "A many-objective test problem for visually examining diversity maintenance behavior in a decision space," in *Proc. 13th Annual Conf. Genetic Evol. Comput. GECCO '11*, 2011, pp. 649–656.

[33] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space," in *Proc. Parallel Problem Solving from Nature PPSN XI*, 2010, pp. 91–100.

[34] A. L. Jaimes and C. A. Coello Coello, "Study of preference relations in many-objective optimization," in *Proc. 11th Annual Conf. Genetic Evol. Comput.*, 2009, pp. 611–618.

[35] A. L. Jaimes, C. A. Coello Coello, H. Aguirre, and K. Tanaka, "Adaptive objective space partitioning using conflict information for many-objective optimization," in *Proc. Evolutionary Multi-Criterion optimization*, 2011, pp. 151–165.

[36] Y. Jin and B. Sendhoff, "A systems approach to evolutionary multiobjective structural optimization and beyond," *IEEE Comput. Intell. Mag.*, vol. 4, pp. 62–76, 2009.

[37] I. Karahan and M. Köksalan, "A territory defining multiobjective evolutionary algorithm and preference incorporation," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 636–664, 2010.

[38] J. R. Kasprzyk, P. M. Reed, B. R. Kirsch, and G. W. Characklis, "Managing population and drought risks using many-objective water portfolio planning under uncertainty," *Water Resour. Res.*, vol. 45, 2009.

[39] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Evolutionary Multi-Criterion Optimization*. Springer Berlin / Heidelberg, 2003, pp. 376–390.

[40] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, Jun. 2000.

[41] ——, "Properties of an adaptive archiving algorithm for storing nondominated vectors," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 100–116, 2003.

[42] ——, "Quantifying the effects of objective space dimension in evolutionary multiobjective optimization," in *Evolutionary Multi-Criterion Optimization*. Springer Berlin / Heidelberg, 2007, pp. 757–771.

[43] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Evolutionary Multi-Criterion Optimization*, 2007, pp. 727–741.

[44] J. W. Kruisselbrink, M. T. Emmerich, T. Bäck, A. Bender, A. P. Ijzerman, and E. Horst, "Combining aggregation with Pareto optimization: A case study in evolutionary molecular design," in *Evolutionary Multi-Criterion Optimization*. Springer Berlin / Heidelberg, 2009, pp. 453–467.

[45] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Proc. Congr. Evol. Comput.*, 2007, pp. 3983–3990.

[46] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, Sep. 2002.

[47] H. Li and Q. Zhang, "Multiobjective optimization problems With complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, 2009.

[48] M. Li and J. Zheng, "Spread assessment for evolutionary multi-objective optimization," in *Proc. 5th Int. Conf. Evol. Multi-Criterion Optimization, EMO '2009*, Nantes, France, 2009, pp. 216–230.

[49] M. Li, J. Zheng, K. Li, Q. Yuan, and R. Shen, "Enhancing diversity for average ranking method in evolutionary many-objective optimization," in *Parallel Problem Solving from Nature, PPSN XI*. Springer Berlin / Heidelberg, 2010, pp. 647–656.

[50] M. Li, J. Zheng, R. Shen, K. Li, and Q. Yuan, "A grid-based fitness strategy for evolutionary many-objective optimization," in *Proc. 12th Annual Conf. Genetic Evol. Comput., GECCO '10*, 2010, pp. 463–470.

[51] K. Maneeratana, K. Boonlong, and N. Chaiyaratana, "Compressed-objective genetic algorithm," in *Parallel Problem Solving from Nature - PPSN IX*. Springer Berlin / Heidelberg, 2006, pp. 473–482.

[52] Z. Michalewicz and D. B. Fogel, *How to solve it: Modern heuristics*. Springer, 2000.

[53] S. Mostaghim and H. Schmeck, "Distance based ranking in many-objective particle swarm optimization," in *Parallel Problem Solving from Nature PPSN X*. Springer Berlin / Heidelberg, 2008, pp. 753–762.

[54] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Evolutionary Multi-Criterion optimization*. Springer, 2001, pp. 82–95.

[55] S. Obayashi and D. Sasaki, "Visualization and data mining of Pareto solutions using self-organizing map," in *Evolutionary Multi-Criterion Optimization*. Springer Berlin / Heidelberg, 2003, pp. 796–809.

[56] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, 2007.

[57] R. C. Purshouse, C. Jalbă, and P. J. Fleming, "Preference-driven co-evolutionary algorithms show promise for many-objective optimisation," in *Evolutionary Multi-Criterion Optimization*, 2011, pp. 136–150.

[58] L. Rachmawati and D. Srinivasain, "Dynamic resizing for grid-based archiving in evolutionary multi objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, 2007, pp. 3975–3982.

[59] H. Sato, H. Aguirre, and K. Tanaka, "Improved S-CDAs using crossover controlling the number of crossed genes for many-objective optimization," in *Proc. 11th Annual Conf. Genetic Evol. Comput. GECCO '11*, 2011, pp. 753–760.

[60] D. Saxena, J. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," KanGAL, Indian Institute of Technology, Tech. Rep. 2010008, 2010.

[61] D. Saxena, Q. Zhang, J. Duro, and A. Tiwari, "Framework for many-objectivet test problems with both simple and complicated Pareto-set shapes," in *Proc. Evolutionary Multi-Criterion optimization 2011*, pp. 197–211.

[62] O. Schütze, A. Lara, and C. A. C. Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 444–455, 2011.

[63] H. K. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 539–556, 2011.

[64] A. Sülflow, N. Drechsler, and R. Drechsler, "Robust multi-objective optimization in high dimensional spaces," in *Evolutionary Multi-Criterion Optimization*. Springer Berlin / Heidelberg, 2007, pp. 715–726.

[65] O. Teytaud, "How entropy-theorems can show that approximating high-dim Pareto-fronts is too hard," in *PPSN-BTP Workshop*, 2006.

[66] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Evolutionary Multi-Criterion Optimization*, 2007, pp. 742–756.

[67] D. J. Walker, R. M. Everson, and J. E. Fieldsend, "Visualisation and ordering of many-objective populations," in *Proc. IEEE Congr. Evol. Comput.*, 2010, pp. 1–8.

[68] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, 2006.

[69] U. K. Wickramasinghe and X. Li, "Using a distance metric to guide PSO algorithms for many-objective optimization," in *11th Annual Conf. Genetic Evol. Comput.*, 2009, pp. 667–674.

[70] G. G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation," *IEEE Trans. Evol. Comput.*, vol. 7, no. 3, pp. 253–274, 2003.

[71] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.

[72] Y. Zhou and J. He, "Convergence analysis of a self-adaptive multi-objective evolutionary algorithm based on grids," *Information Processing Letters*, vol. 104, no. 4, pp. 117–122, 2007.

[73] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization,"

in *Evolutionary Methods for Design, Optimisation and Control.*, 2002, pp. 95–100.

[74] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature - PPSN VIII*. Springer Berlin / Heidelberg, 2004, pp. 832–842.

[75] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Trans. Syst., Man, and Cybern., Part B: Cybern.*, vol. 38, no. 5, pp. 1402–1412, 2008.