Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

ETEA: A Euclidean Minimum Spanning Tree-Based Evolutionary Algorithm for Multi-Objective Optimization

Miqing Li miqing.li@brunel.ac.uk
Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, U.K.

Shengxiang Yang syang@dmu.ac.uk
School of Computer Science and Informatics, De Montfort University, Leicester, LE1 9BH, U.K.

Jinhua Zheng jhzheng@xtu.edu.cn
Institute of Information Engineering, Xiangtan University, Xiangtan, 411105, China

Xiaohui Liu xiaohui.liu@brunel.ac.uk
Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, UB8 3PH, U.K.

doi:10.1162/EVCO_a_00106

Abstract

The Euclidean minimum spanning tree (EMST), widely used in a variety of domains, is a minimum spanning tree of a set of points in space where the edge weight between each pair of points is their Euclidean distance. Since the generation of an EMST is entirely determined by the Euclidean distance between solutions (points), the properties of EMSTs have a close relation with the distribution and position information of solutions. This paper explores the properties of EMSTs and proposes an EMST-based evolutionary algorithm (ETEA) to solve multi-objective optimization problems (MOPs). Unlike most EMO algorithms that focus on the Pareto dominance relation, the proposed algorithm mainly considers distance-based measures to evaluate and compare individuals during the evolutionary search. Specifically, in ETEA, four strategies are introduced: (1) An EMST-based crowding distance (ETCD) is presented to estimate the density of individuals in the population; (2) A distance comparison approach incorporating ETCD is used to assign the fitness value for individuals; (3) A fitness adjustment technique is designed to avoid the partial overcrowding in environmental selection; (4) Three diversity indicators—the minimum edge, degree, and ETCD—with regard to EMSTs are applied to determine the survival of individuals in archive truncation. From a series of extensive experiments on 32 test instances with different characteristics, ETEA is found to be competitive against five state-of-the-art algorithms and its predecessor in providing a good balance among convergence, uniformity, and spread.

Keywords

Multi-objective optimization, evolutionary algorithms, Euclidean minimum spanning tree, density estimation, fitness assignment, fitness adjustment, archive truncation.

1 Introduction

Many real-world problems involve simultaneous optimization of several competing objectives. In these multi-objective optimization problems (MOPs), there is usually no

Manuscript received: June 4, 2011; revised: April 28, 2012, January 11, 2013, and April 18, 2013; accepted: May 6, 2013.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

single optimal solution, but rather a set of alternative solutions, called the Pareto set, due to the conflicting nature of the objectives. In the absence of any further information, the decision-makers usually require an approximation of the Pareto set for making their final choice.

Over the past few years, evolutionary algorithms (EAs) have been gaining increasing attention among researchers and practitioners to solve MOPs (Coello et al., 2007; Deb, 2001; Branke et al., 2008). One main advantage of EAs is that they have low requirements on the problem characteristics (e.g., nonconvexity, discontinuity, nonlinear constraint, and multimodality), and objectives can be easily added, removed, or modified. Moreover, due to the fact that they act on a set of candidates, EAs are suitable for generating a Pareto set approximation in a single run.

As a consequence, numerous effective evolutionary multi-objective optimization (EMO) algorithms have been proposed, such as the non-dominated sorting genetic algorithm II (NSGA-II; Deb et al., 2002), strength Pareto evolutionary algorithm 2 (SPEA2; Zitzler et al., 2002), Pareto-based evolution strategy (PAES; Knowles and Corne, 2000), indicator-based evolutionary algorithm (IBEA; Zitzler and Künzli, 2004), ϵ-dominance (Laumanns et al., 2002) based multi-objective evolutionary algorithm (ϵ-MOEA; Deb, Mohan, et al., 2005), multi-objective covariance matrix adaptation evolution strategy (MO-CMA-ES; Igel, Hansen, et al., 2007), S metric selection evolutionary multi-objective optimization algorithm (SMS-EMOA; Beume et al., 2007), and decomposition-based multi-objective evolutionary algorithm (MOEA/D; Zhang and Li, 2007), some of which are applied to various problem domains (see Fonseca and Fleming, 1995; Coello and Lamont, 2004; Tan et al., 2005; Abraham et al., 2005; Jin, 2006; Bui and Alam, 2008; Wang et al., 2010; Teo and Abbass, 2004; Friedrich et al., 2010). Generally speaking, these algorithms share the three common goals—minimizing the distance to the optimal front, maintaining the uniform distribution, and extending the distribution range along the optimal front.

In general, EMO algorithms, based on their selection mechanisms, can be divided into three groups: Pareto-based algorithms, aggregation-based algorithms, and indicator-based algorithms (Coello, 2011; Wagner et al., 2007).

The main idea of Pareto-based algorithms is to compare individuals of a population based on their Pareto dominance relation and distribution. The Pareto dominance relation is used to distinguish individuals in terms of convergence, and the distribution is used to maintain the diversity of individuals in the population. Many effective EMO algorithms belong to this group. Among them, NSGA-II (Deb et al., 2002) and SPEA2 (Zitzler et al., 2002) are two representative algorithms.

In aggregation-based algorithms, the objectives are normally aggregated in some form (using either linear or nonlinear schemes), such that a single scalar value is generated. This scalar value is used as the fitness of the algorithm. In comparison with the algorithms in other groups, aggregation-based algorithms require a priori definition of relations among objective functions. As the earliest multi-objective optimization method that can be traced back to the middle of the last century (Kuhn and Tucker, 1951), the aggregation-based approach has become popular again in recent years, partially due to the appearance of an effective algorithm, MOEA/D (Zhang and Li, 2007).

The basic idea behind indicator-based algorithms is to employ a performance indicator to select individuals. One important characteristic of indicator-based algorithms is that in contrast to Pareto-based algorithms which compare individuals using two criteria (i.e., Pareto dominance relation and distribution), these algorithms adopt a single indicator to optimize a desired property of the evolutionary population. The algorithm IBEA (Zitzler and Künzli, 2004) is a pioneer in this group. Recently, some algorithms

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

in this group, such as SMS-EMOA (Beume et al., 2007) and HypE (Bader and Zitzler, 2011), have been found to be promising in solving many-objective optimization problems (Wagner et al., 2007; Bader and Zitzler, 2011; Li et al., 2013).

This paper focuses on Pareto-based EMO algorithms. In these algorithms, the convergence of individuals in the population is estimated according to the Pareto dominance relation based fitness strategies, such as the dominance count (Fonseca and Fleming, 1995), strength (Zitzler et al., 2002), and dominance rank (Deb et al., 2002). However, such estimation depending fully on the Pareto dominance relation may lead to the existence of a large amount of incomparable individuals in the population due to the lack of the quantitative measure (see Farina and Amato, 2003; Ishibuchi et al., 2008; Yang et al., 2013). On the other hand, with respect to diversity, most algorithms only consider the crowding degree of individuals, but ignore the position of individuals in the population. In fact, the position of individuals also has an important influence on diversity since the uniformity and spread of the entire population need to be maintained (a detailed explanation is given in the latter part of Section 3.1).

In this paper, we develop a Euclidean minimum spanning tree (EMST) based EA (ETEA) to address the above issues. The aim of the paper is to employ the characteristics of EMSTs and the distance relation among individuals to balance the convergence, uniformity, and spread of the population during the evolutionary search. To this end, firstly, an EMST-based density estimator is proposed to measure the crowding degree and position of individuals in the population. Secondly, two distance-based measures incorporating the Pareto dominance relation are used to compare individuals in fitness assignment and environmental selection. Finally, three EMST-related indicators are applied to maintain the archive set when the number of non-dominated individuals exceeds the size of the set.

The EMST is a minimum spanning tree of a set of points in the space, where the weight of the edge between each pair of points is their Euclidean distance. In other words, an EMST connects a set of points in the space using lines in order to obtain the minimized total length of all the lines and reach any point from any others through the exclusive lines. EMSTs can be applied in a wide variety of domains, such as the network, piping, Euclidean traveling salesman problems, among others (Lee, 1999; Bansal and Ghanshani, 2006; Wieland et al., 2007; Šeda, 2008).

Since the generation of an EMST is entirely determined by the Euclidean distance between solutions (points), some properties in EMSTs generally have a close relationship with the distribution and position information of the solutions. For example,

- Solutions which are distributed in more crowded regions have shorter edges;
- The boundary solutions are often of low node degrees, yet some bridge-like solutions have high node degrees;
- The line between an individual and its neighbor whose orientation is different from others may have a higher likelihood of becoming an edge of the EMST;
- The EMST which is constructed by a non-dominated set in the 2-dimensional space degenerates into linear structure.

In this paper, we will employ these properties to deal with MOPs.

As a first attempt to capture and utilize the properties of EMSTs in EMO, we have recently developed a fitness assignment strategy and a diversity maintenance approach in Li et al. (2008). In view of encouraging experimental results of these preliminary

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

studies, this paper conducts a further and thorough investigation along this line. In comparison with the previous work, the main contributions of this paper are summarized as follows.

1. An elaborate fitness assignment scheme is designed, which takes a distance comparison relation between non-dominated individuals and dominated ones into account, instead of the simple distance evaluation in Li et al. (2008).
2. A fitness adjustment technique is introduced to avoid partial overcrowding by penalizing the individuals once their neighbors have been picked out during the environmental selection process.
3. An improved population truncation method is proposed to preserve the boundary solutions as well as to eliminate crowded solutions in the archive.
4. Systematic experiments are carried out to compare ETEA with five state-of-the-art algorithms on 32 test problems; only NSGA-II and SPEA2 were used to validate the proposed algorithm on a few problems in Li et al. (2008). In addition, this paper also contains a comparative study between ETEA and its predecessor, an analytical and empirical study of computational cost, and an investigation of different parts of the proposed algorithm.

The rest of this paper is organized as follows. In Section 2, relevant notation and definitions are reviewed. Section 3 is devoted to the description of the proposed algorithm. Section 4 presents the algorithm settings, test functions, and performance metrics. Experimental results are presented and analyzed in Section 5. Finally, Section 6 concludes the paper and presents future work.

2 Definitions and Terminology

The concepts of Pareto optimality have been well understood in the literature. This section will introduce notation closely related to our work, such as extreme solutions and boundary solutions.

Without loss of generality, we suppose that an arbitrary MOP consists of m objectives, which are all to be minimized and equally preferable. A solution to this MOP can be described in terms of a decision vector $(x_1, x_2, \ldots, x_n)$ in the decision space $\boldsymbol{X}$. A function $\boldsymbol{F} : \boldsymbol{X} \to \boldsymbol{Y}$ evaluates the quality of a specific solution by assigning it an objective vector $[f_1(x_1, x_2, \ldots, x_n), \ldots, f_m(x_1, x_2, \ldots, x_n)]$ in the objective space $\boldsymbol{Y}$.

Pareto optimality is defined by using the concept of dominance. Given two decision vectors a and b, a is said to *dominate* b (denoted as $a \prec b$), iff a is at least as good as b in all objectives and better in at least one objective. Accordingly, those decision vectors that are not dominated by any other vectors are denoted as *Pareto optimal solutions*. In general, the set of optimal solutions in the decision space is denoted as the *Pareto set*, and the corresponding set of objective vectors as the *Pareto front*. Unfortunately, it is often infeasible to obtain the Pareto set, and it is only hoped to find a good approximation of the set. Usually, we consider the *nondominated set* found in one run as the approximation.

Although the solutions in a non-dominated set are incomparable with each other on the basis of the Pareto dominance concept, their positions that affect the distribution range of the set can be well distinguished. Several concepts about the range of a non-dominated set are introduced as follows.

DEFINITION 1 (EXTREME SOLUTIONS): *The solutions in a non-dominated set have the maximum value for at least one objective.*

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

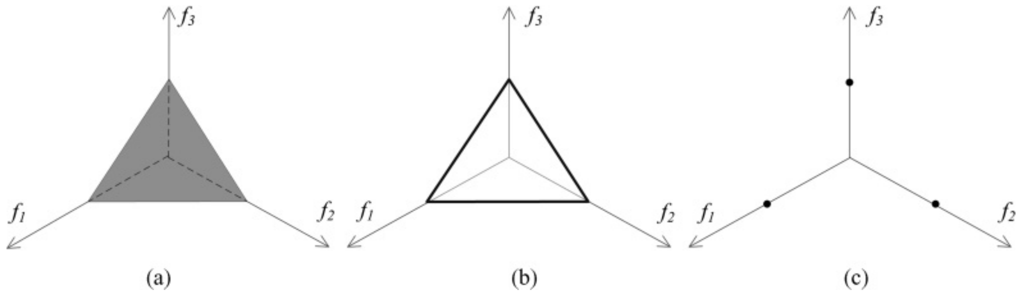

Figure 1: A tri-objective example of boundary solutions and extreme solutions of a Pareto front: (a) Pareto front, (b) boundary solutions, and (c) extreme solutions.

The extreme solutions, which are used in numerous diversity maintenance strategies and performance assessment techniques, can partly reflect the extent of a non-dominated set. Especially for bi-objective problems, the extreme solutions play a decisive role in the distribution range. The greater the distance between two extreme solutions, the wider the distribution range of the non-dominated set. However, the extreme solutions fail to provide enough information to report the range of solutions for problems with more than two objectives. To this end, a concept of *boundary solutions* (Li and Zheng, 2009) has been presented to overcome this shortcoming. In order to define the boundary solutions, a comparison relation between individuals, called *beyond*, is first introduced as follows.

DEFINITION 2 (BEYOND): *A vector a is said to* beyond *a vector b in the objective space* $(f_1, f_2, \ldots, f_m)$, *if* $f_i(a) \geq f_i(b)$ *for all* $i \in \{1, 2, \ldots, m\}$ *and* $f_j(a) > f_j(b)$ *for some* $j \in \{1, 2, \ldots, m\}$.

Note that the definition of beyond is equal to that of the Pareto dominance relation regarding a maximization MOP. In the following, the definition of the boundary solutions in a non-dominated set is given according to the beyond relation between solutions in the set.

DEFINITION 3 (BOUNDARY SOLUTIONS IN THE OBJECTIVE SPACE $(f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_m)$ AND BOUNDARY SOLUTIONS): *A vector a in a non-dominated set S is considered as a boundary solution in the objective space* $(f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_m)$ *(denoted as* BS_i*), if a is not beyond by any member of S for the subset* $\{f_1, \ldots, f_{i-1}, f_{i+1}, \ldots, f_m\}$ *of all the objectives. A vector a is said to be a boundary solution of S if a is one of the vectors in* $BS_1 \cup BS_2 \cup \ldots \cup BS_{m-1} \cup BS_m$.

The boundary solutions of a non-dominated set entirely determine its range. They are significantly different from extreme solutions, despite the fact that boundary solutions in general include extreme solutions and are even equal to extreme solutions on bi-objective problems. Figure 1 gives an example of boundary solutions and extreme solutions. A more detailed description and analysis can be found in Li and Zheng (2009).

3 Description of the Proposed Algorithm

ETEA is an EMO algorithm which utilizes the properties of EMSTs to solve MOPs. In this section, we first present the main loop of ETEA and a density estimator based on EMSTs. Then, we describe the fitness assignment process. Next, we introduce the fitness

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Algorithm 1 Mainloop

Require: L (population size), N (archive size)
1. Generate an initial population P_0 and create an archive set Q_0 and temporary set R_0. Set $t \leftarrow 0$
2. **while** termination criterion not fulfilled **do**
3. $\quad R_t \leftarrow Copy(P_t, Q_t)$ /* Copy all individuals in P_t and Q_t to R_t */
4. $\quad Fitness\_assignment(R_t)$
5. $\quad Q_{t+1} \leftarrow Elitism\_selection(R_t)$ /* Copy non-dominated individuals in R_t to Q_{t+1} */
6. $\quad$ **if** $|Q_{t+1}| < N$ **then**
7. $\quad\quad Q_{t+1} \leftarrow Fitness\_adjustment(R_t) \cup Q_{t+1}$
   /* Put the best dominated individuals into Q_{t+1} by the fitness adjustment process */
8. $\quad$ **else**
9. $\quad\quad Q_{t+1} \leftarrow Archive\_truncation(Q_{t+1})$ /* Truncate archive Q_{t+1} */
10. $\quad$ **end if**
11. $\quad P_{t+1} \leftarrow Mating\_selection(Q_{t+1})$ /* Perform mating selection on Q_{t+1} */
12. $\quad P_{t+1} \leftarrow Variation(P_{t+1})$ /* Apply variation operators to P_{t+1} */
13. $\quad t \leftarrow t + 1$
14. **end while**
15. **return** Q_t

adjustment technique in environmental selection. Finally, a truncation strategy is given to maintain diversity in the archive.

3.1 Main Loop and Density Estimation

The main loop of ETEA is given in Algorithm 1. Clearly, the basic procedure of the algorithm is similar to general generational EMO algorithms except that a fitness adjustment strategy is added in environmental selection (shown in line 7). Most of the generational EMO algorithms (such as NSGA-II and SPEA2) directly select the best dominated individuals according to fitness information when the non-dominated individuals are not enough to fill the archive. A shortcoming of this strategy is that it may lead to the loss of diversity since neighboring individuals often have similar fitness values. The specific process of fitness adjustment will be described in Section 3.3. In addition, it should be pointed out that this paper only focuses on fitness assignment (line 4) and environmental selection which consists of elitism selection, fitness adjustment, and archive truncation (lines 5–10). In other words, mating selection and variation schemes in ETEA are not determined and can be freely selected by users. In the following, we present a density estimator which guides the search at different parts of the algorithm.

Most EMO algorithms try to maintain diversity by incorporating density information into the selection process (see Horoba and Neumann, 2010): the higher the density of the surrounding area of an individual in a population, the lower the chance of the individual being selected. In other words, density estimation is needed in EMO algorithms to encourage uniform distribution of individuals over the current trade-off surface. In this paper, we employ the edges of an individual (node) in the EMST to estimate its distribution. An estimator, called the *Euclidean minimum spanning tree crowding distance*, is given here.

DEFINITION 4 (EUCLIDEAN MINIMUM SPANNING TREE CROWDING DISTANCE): *Let T be a Euclidean minimum spanning tree of a solution set P. For an individual X of P, let $Y_i (i = 1, \ldots, d)$ denote the individuals sharing an edge with X, where d is the number of edges attached to X (i.e., the degree of node X in the EMST; e.g., for node* **D** *in Figure 2, $d = 3$), and L_{XY_i} denote the length (weight) of the edge $XY_i (i = 1, \ldots, d)$, i.e., the Euclidean distance between*

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

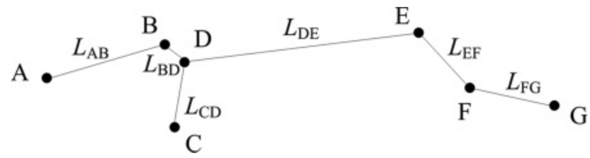

Figure 2: An EMST of the set {**A**, **B**, **C**, **D**, **E**, **F**, **G**}, where L_{XY} denotes the length of the edge between solutions X and Y.

individuals X and Y_i. The Euclidean minimum spanning tree crowding distance (ETCD) of X is defined as follows:

$$ETCD(X) = \left(\sum_{i=1}^{d} {L_{XY_i}}^{0.5} / d \right)^2 \tag{1}$$

Clearly, the ETCD of an individual is the kth power mean of the length of all its edges, where k is equal to 0.5. For instance, in Figure 2, the density estimator of individual **F** is determined by $L_{\mathbf{EF}}$ and $L_{\mathbf{FG}}$, and its ETCD is the 0.5th power mean of them. Here, assigning k the value 0.5 is a rough setting in order to obtain a tradeoff among the effects of the neighbors of X with different distances. If k is set to 1.0 (i.e., ETCD is the arithmetic mean of edge weights), all neighbors of X will have the same contribution to the density of X no matter how far they are from X, which partly hinders the development of uniformity of the population (see the example in the third observation of ETCD in the list following the next paragraph). Therefore, a value of k lower than 1.0 may be suitable for emphasizing the effect of the closer neighbors. However, when k approximates 0, almost only the closest neighbor will contribute to the density of X, which apparently ignores the effects of the other neighbors. Therefore, we simply set k to the middle value between 0 and 1. In fact, other values between 0 and 1 can also be adopted as long as they are away from the boundaries 0 and 1.

Similar to other density estimators, the effectiveness and characteristics of ETCD rely heavily on the properties of the assessment technique, since different techniques will lead to different judgments on density estimation. From the calculation of the proposed estimator, we can draw some observations as follows.

1. In accordance with the greediness of the procedure of constructing an EMST, the edge between an individual and its closest neighbor (i.e., the individual which has the shortest Euclidean distance to it) belongs to the EMST. Accordingly, from the second shortest edge to others, they generally have a decreasing chance to become a component of the EMST.

2. According to the connectivity of an EMST, the line between an individual and its neighbor whose orientation is different from others may have a higher likelihood of becoming an edge of the EMST. For example, in Figure 2, for individual **B** and its neighbors **A** and **C**, the line between **B** and **A** belongs to EMST in contrast to the line between **B** and **C**, although the former is longer than the latter. This is because relative to **B**, **A** has a different orientation against other neighbors around **B**; yet there exists a closer neighbor (**D**) who has a similar orientation to **C** with regard to **B**. Moreover, the second behavior derived from the connectivity of an EMST is that some bridge-like individuals that connect two clusters of individuals have higher ETCD values. For example, individuals **D** and **E** in Figure 2 may be regarded as intermediate individuals joining two

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

clusters ({**A**, **B**, **C**, **D**} and {**E**, **F**, **G**}). For individual **D**, clearly, a relatively higher ETCD value is obtained since $L_{\mathbf{DE}}$ is included in the calculation of the estimator. In summary, from the above discussion, it becomes clear that the proposed estimator prefers the individuals which can be regarded as an intermediate connection for other members in the population. This phenomenon seems to be consistent with the target of advancing the uniformity of distribution. This is because these intermediates, in contrast to their neighbors, are often located closer to other individuals (or clusters), and thus their offspring have a higher likelihood of filling the empty areas between them and those individuals (or clusters).

3. Note that the definition of ETCD is slightly different from that of the density estimator in Li et al. (2008). In Li et al. (2008), the density estimator was defined by calculating the arithmetic mean of edge weights. Here, the 0.5th power mean is used to replace the arithmetic mean for improving uniformity. For example, consider individuals **B** and **F** in Figure 2 regarding the two density estimators, and assume $L_{\mathbf{AB}} = 9.0$, $L_{\mathbf{BD}} = 1.0$, $L_{\mathbf{EF}} = 5.0$, and $L_{\mathbf{FG}} = 5.0$. Clearly, according to Li et al. (2008), the estimation value of **B** (5.0) is equal to that of **F** (5.0); yet for ETCD, **B** performs worse than **F** (4.0 against 5.0).

The main difference between ETCD and other density estimators is that ETCD not only reflects the crowding degree but partly implies the relative orientation and position information of individuals. Yet most of the existing density estimators (such as the niche techniques, Horn et al., 1994; Tan et al., 2001; Shir et al., 2010; crowding distance, Deb et al., 2002; Nebro et al., 2008; *k*th nearest neighbor, Zitzler et al., 2002; Elhossini et al., 2010; and grid crowding degree, Corne et al., 2001; Yen and Lu, 2003; Li et al., 2010) only evaluate the density information of individuals. Although these strategies seem to be reasonable, they may be imprecise due to the influence of individuals' position: the individuals located on or near the border of a population usually have a lower crowding degree; some bridge-like individuals, which are of great service to uniformity, may be distributed in the region with a high crowding degree. For example, considering individual **D** in Figure 2, it may be assigned a high density value by some estimators (e.g., the niche techniques, crowding distance, *k*th nearest neighbor, and grid degree), thus being eliminated early. However, as previously discussed, individual **D** is important in the context of maintaining uniformity and can be regarded as an intermediate individual connecting two clusters {**A**, **B**, **C**, **D**} and {**E**, **F**, **G**}.

3.2 Fitness Assignment

In order to evolve a population toward the optimum as well as to diversify its individuals uniformly along the obtained trade-off surface, the fitness value of individuals should be assigned to reflect both convergence and diversity accordingly. At present, most studies on fitness assignment mainly focus on the issue of the Pareto dominance relation, such as the dominance count, strength, dominance rank, and others (Bosman and Thierens, 2003; Li, 2003; Gong et al., 2008). In this paper, we prefer the distance from individuals to the obtained trade-off surface. We consider the distance differences among some specific individuals and record the successful counts of them (called the distance count here). In detail, with respect to the distance count, we distinguish between non-dominated individuals and dominated ones. For a non-dominated individual, its distance count is assigned to zero. For a dominated individual, denoted as individual i, the non-dominated individual j which dominates and is the closest to i

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

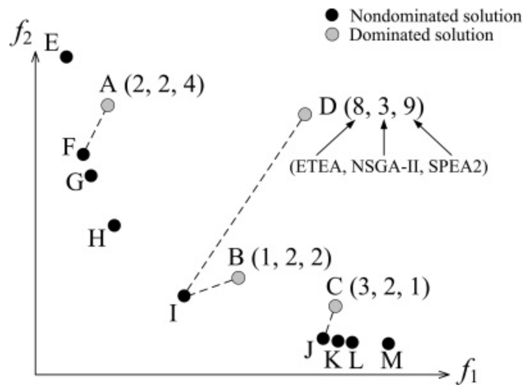

Figure 3: Comparison of fitness assignment strategies for a minimization bi-objective problem. The numbers in the parentheses associated with the dominated solutions correspond to the distance count, dominance rank, and strength in ETEA, NSGA-II, and SPEA2, respectively. The dashed lines connect the dominated solutions to their corresponding non-dominated solutions in the distance count calculation.

is first selected. Then, the distance count of i is determined by the total number of the non-dominated individuals whose distance from j is shorter than the distance between i and j:

$$D(i) = \begin{cases} \left|\{k|L_{jk} < L_{ij} \wedge k \in NDS, k \neq j\}\right| + 1, & i \in DS \\ 0, & i \in NDS \end{cases} \tag{2}$$

where

$$j \in NDS \wedge j \prec i \wedge (\neg\exists r \in NDS, r \prec i \wedge L_{ir} < L_{ij}) \tag{3}$$

where $\left|\cdot\right|$ denotes the cardinality of a set, L_{ij} implies the distance from i to j, and DS and NDS represent the set of dominated and non-dominated solutions, respectively. Here, the distance count is minimized, and for dominated individuals, it is penalized by adding one in order to guarantee that they have a larger value than non-dominated individuals. For example, let us consider dominated individual **A** in Figure 3. First, individual **F** is selected since it is the non-dominated individual which dominates and is the closest to **A**. Then, we look for non-dominated individuals which can contribute to the distance count of **A**. Here, only individual **G** is qualified, considering that its distance from **F** is shorter than the distance between **A** and **F**. Thus, the distance count of **A** is $\left|\{\mathbf{G}\}\right| + 1 = 2$. To better understand the characteristics of this scheme, an example of the distance count in comparison with two well-known strategies (the dominance rank and strength) in NSGA-II and SPEA2 is illustrated in Figure 3.

Clearly, the distance count of dominated individuals is mainly determined by two factors: (1) their distance from the non-dominated front, and (2) the distance between the corresponding non-dominated individual and other ones. An individual with a poor distance count means that it is far away from the non-dominated front, or the non-dominated individual in the population who is the closest to and dominates it is located in a crowded region. In Figure 3, individual **D** illustrates the first factor: it is located far away from the non-dominated front, thereby obtaining a high distance count; on the other hand, individual **C** provides an example for the second factor: since its corresponding non-dominated individual is located in a crowded region, **C** is assigned

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

a relatively high distance count value even if it approximates the non-dominated front. However, the other two strategies (depending on dominance information) are not able to effectively distinguish this case.

It is worthwhile to mention that a significant difference between ETEA and other strategies is that ETEA places more emphasis on the distribution of non-dominated individuals, since its fitness strategy takes into account the distance measurement among individuals. Actually, non-dominated individuals play a crucial role in the selection process of EMO. The non-dominated front that is composed of these individuals can largely determine the search direction and reflect the evolution bias in distinct areas. Therefore, a non-dominated front with uniformly and widely distributed individuals is considerably important and able to drive the whole population toward the desired direction. Naturally, some dominated individuals who have a high likelihood of achieving this target (i.e., they are located near the sparse regions of the non-dominated front) should be assigned better fitness values even if they are dominated by some individuals, such as individual **A** in Figure 3.

Although the distance count provides elaborate preference information for dominated individuals, it fails when most individuals in the population do not dominate each other because it is equal to zero for all non-dominated individuals. In addition, the density information of each individual in the population cannot also be directly reflected according to the distance count. Therefore, we incorporate ETCD into the fitness in order to discriminate the individuals who have identical distance count as well as to provide a density indicator for each individual. Here, we take the inverse of ETCD in accordance with the minimization of the distance count value. Accordingly, the fitness of individual i is defined as follows:

$$F(i) = D(i) + \frac{1}{\mathrm{ETCD}(i) + 1} \tag{4}$$

In the fraction of Equation (4), one is added to the denominator to ensure that its value is greater than zero and smaller than or equal to one. As a result, the fitness for non-dominated individuals is within the range of (0, 1], and for dominated individuals larger than one.

3.3 Fitness Adjustment

Mating selection and environmental selection are two indispensable parts of an EMO algorithm. Although both of them are based on fitness information of individuals, they are, in principle, fully independent of each other. Mating selection aims at picking promising individuals for variation and is usually performed in a random way. In contrast, environmental selection determines which of the previously stored individuals and the newly created ones are kept in the archive (Zitzler et al., 2004).

Unfortunately, most current EMO algorithms, such as NSGA-II and SPEA2, do not distinguish this difference and often directly perform the selection operation according to the straightforward fitness rank of individuals. In fact, in contrast to mating selection, where the directly-selected way seems to be reasonable due to the randomness of the selection, the environmental selection based on the straightforward fitness rank may reduce the diversity of the archive because of the deterministic way in which individuals move into the archive, ordered by their level of fitness. Since the fitness value of individuals depends on their position compared with other individuals in the population, those individuals that are closely located often have similar values. Therefore, it is very likely that they are eliminated or preserved simultaneously, which may bring about individuals crowded in some regions yet produce vacancies in other regions.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Algorithm 2 Fitness_adjustment(R)

Require: Q (non-dominated set), N (archive size)
1. Generate an empty set S for storing the best dominated individuals and create an empty temporary set T for storing their neighbors. Set $Select\_num \leftarrow N - |Q|$
2. **while** $|S| < Select\_num$ **do**
3. $\quad p \leftarrow Findout\_best(R)$
$\quad\quad$ /* Find out the best dominated individual p (i.e., p has the minimum fitness value) */
4. $\quad T \leftarrow Findout\_neighbor(R, p)$
$\quad\quad$ /* Find out the neighboring dominated individuals of p in R */
5. $\quad Sort(T, p)$
$\quad\quad$ /* Sort all individuals in T with decreasing order according to the distance from p */
6. $\quad$ **for all** $q_i \in T, i = 1, ..., |T|$ **do**
7. $\quad\quad F(q_i) \leftarrow F(q_i) + i$ $\quad$ /* Adjust the fitness value of the ith individual in T */
8. $\quad$ **end for**
9. $\quad S \leftarrow S \cup \{p\}$ $\quad$ /* Add individual p into S */
10. $\quad R \leftarrow R \backslash \{p\}$ $\quad$ /* Remove individual p from R */
11. **end while**
12. **return** S

In this study, we propose a fitness adjustment strategy in environmental selection. The individuals are penalized once their neighbors have been selected into the archive. Specifically, we consider the circle centered at the selected individual as its neighborhood whose range is determined by the distance between it and the non-dominated front. For individuals in the neighborhood, a hierarchical fitness penalty is executed according to their distance from the center individual. It should be noted that this adjustment occurs when the non-dominated individuals are not enough to fill the archive, and it only aims at the dominated individuals. Algorithm 2 gives the detailed procedure of this fitness adjustment strategy.

In Algorithm 2, Function $Findout\_neighbor(R, p)$ (line 4) is designed to find out the neighbors of the current best dominated individual p in population R. The neighborhood radius is defined by the distance from the center individual (i.e., the selected individual) to its nearest non-dominated individual who dominates it. Lines 6–8 of the algorithm inflict a fitness penalty on the neighbors of the selected individual. The penalty degree of individuals relies on the crowding degree reflected by the total number of individuals in the neighborhood as well as on the distance between them and the center. Therefore, a more crowded neighborhood leads to a higher overall penalty; and for each individual, the further it is from the center, the milder the penalty.

An example of fitness adjustment is illustrated in Figure 4. It is clear that the penalty mechanism in ETEA largely avoids crowding in the archive, because once an individual is picked out, its neighbors will be penalized (see Figure 4(a)–(d)). However, the selection strategies in NSGA-II and SPEA2, which are directly performed according to the fitness of individuals, reduce the diversity to some extent. Specifically, for NSGA-II, since individuals **A–F** have the same dominance rank, the three most crowded individuals **C**, **D**, and **E** will be eliminated. As to SPEA2, since the calculation of fitness of an individual is based on the strength of the individuals that dominate it, individuals **B**, **C**, and **F**, which are dominated by the individuals that have larger strength values, will be eliminated.

3.4 Archive Truncation

As described in Algorithm 1, the first step in environmental selection is to copy all non-dominated individuals into the archive. If there are still a certain number of available

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

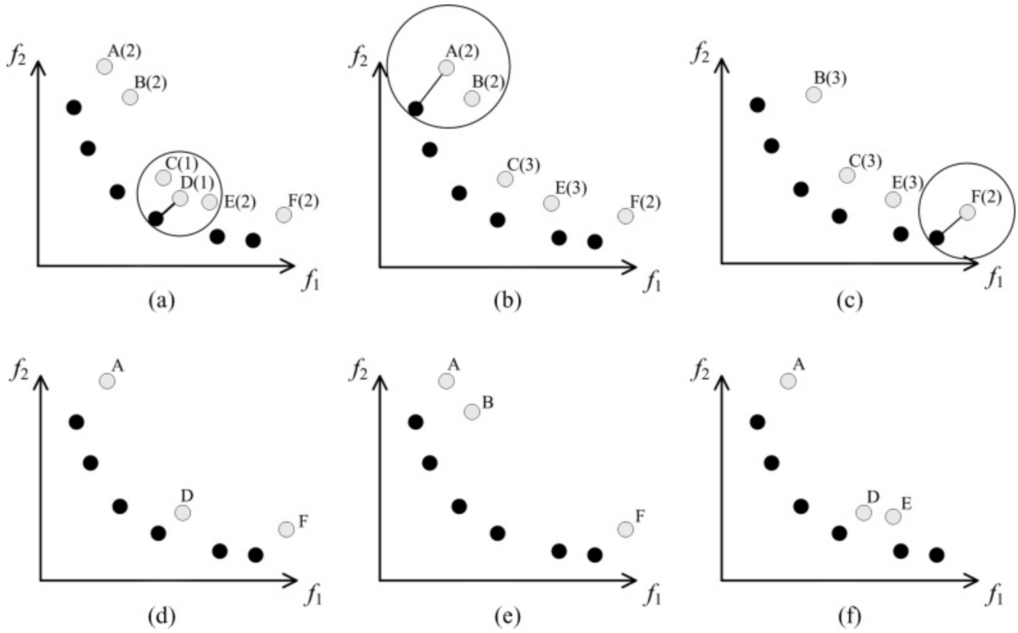

Figure 4: A scenario of the fitness adjustment procedure in ETEA and its result compared with that of NSGA-II and SPEA2. (a) Original set R. (b) **D**-eliminated set R. (c) **A**-eliminated set R. (d) Final archive of ETEA. (e) Final archive of NSGA-II. (f) Final archive of SPEA2. Where $Select\_num = 3$. **A**, **B**, **C**, **D**, **E**, and **F** are the candidate dominated individuals. (a)–(c) are the fitness adjustment procedure in environmental selection of ETEA; (d)–(f) are the final archive results by the environmental selection process of the three algorithms. The number in the parentheses associated with each candidate individual means the integral part of the fitness value (i.e., the distance count) in ETEA. The circle corresponds to the neighborhood of the current best dominated individual.

slots in the archive, some best dominated individuals will fill the archive according to the fitness adjustment strategy in the previous section. If the size of these non-dominated individuals exceeds the upper bound of the archive, an archive truncation procedure is activated to remove some individuals for obtaining a representative archive. However, obtaining a representative archive is not a trivial task, since both properties of distribution (i.e., uniformity and spread) are supposed to be taken into account. On the one hand, as a non-dominated front can be a convex, non-convex, disconnected, or piecewise continuous hypersurface, the difficulty may arise regarding how to maintain its proper distribution shape. On the other hand, the boundary effect will emerge when the uniformity of a non-dominated set is considered (Farhang-Mehr and Azarm, 2002). The number of the neighbors of outer individuals is generally less than that of inner ones, even if they have a higher crowding degree. This may result in a misleading estimation of individuals' density. In addition, the reasonable integration of both properties (uniformity and spread) into one truncation method is also a noticeable issue. The improvement of performance at one point should not cause a simultaneous degradation at the other point.

In this study, we propose an archive truncation strategy by employing the EMST to maintain uniformity and spread. The pseudocode is given in Algorithm 3. The main

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Algorithm 3 $Archive\_truncation(Q)$

Require: N (archive size)
1. $Calculate\_distance(Q)$ /* Calculate the Euclidean distance between individuals in Q */
2. **while** $|Q| > N$ **do**
3. $Construct\_EMST(Q)$ /* Construct an EMST using the Prim algorithm for all individuals in Q according to the Euclidean distance of individuals, and meanwhile calculate the ETCD and degree of each individual, denoted as $ETCD(i)$ and $De(i)$ for individual i */
4. $L_{pq} \leftarrow Find\_minimum(Q)$ /* Find out the edge with the minimum weight in the EMST, denoted as L_{pq}, where p and q are a pair of endpoints of L_{pq} */
/* If the degree of one individual is equal to 1, then eliminate the other */
5. **if** $De(p) = 1$ **or** $De(q) = 1$ **then**
6. **if** $De(p) = 1$ **then**
7. $Q \leftarrow Q \setminus \{q\}$
8. **else**
9. $Q \leftarrow Q \setminus \{p\}$
10. **end if**
/* If the degree of both individuals is greater than 1, then compare their ETCD */
11. **else**
12. $ETCD'(p) \leftarrow ((ETCD(p)^{0.5} \times De(p) - {L_{pq}}^{0.5}) / (De(p) - 1))^2$ /* Modify the original ETCD of p, i.e., remove L_{pq} and recalculate the ETCD for the remaining edges of p */
13. $ETCD'(q) \leftarrow ((ETCD(q)^{0.5} \times De(q) - {L_{pq}}^{0.5}) / (De(q) - 1))^2$
14. **if** $ETCD'(p) < ETCD'(q)$ **then**
15. $Q \leftarrow Q \setminus \{p\}$
16. **else**
17. $Q \leftarrow Q \setminus \{q\}$
18. **end if**
19. **end if**
20. **end while**
21. **return** Q

procedure of the truncation includes three steps. Firstly, an edge with the minimum weight is found in the EMST (line 4), and the two endpoints of the edge are regarded as the candidate individuals to be considered. Secondly, the degree property is introduced to determine their survival (lines 5–10). If the degree value of one candidate is equal to one, the other candidate is eliminated (according to the connectivity of an EMST, there should not be two candidates whose degree is one unless the size of the set is equal to two). Finally, if the degree values of both candidates are larger than one, the candidate with a higher ETCD value is preferable (lines 11–18). Note that the original ETCD of candidates has been slightly modified here: the edge with the minimum weight is removed in the calculation of ETCD. A detailed analysis with regard to this modification will be presented in the last part of this section.

Figure 5 shows an illustration of the truncation procedure for a tri-objective non-dominated set. Firstly, an EMST of the original non-dominated set is generated, and then the shortest edge $L_{\mathbf{AB}}$ is found. Individual **B** is eliminated since the degree of **A** is equal to one. And again a new EMST of the remaining individuals is generated, and similarly candidates **C** and **E** are found. **E** is eliminated because (1) the degrees of both candidates are greater than one, and (2) the modified ETCD of **C** is larger than that of **E** (i.e., the length of edge $L_{\mathbf{CA}}$ is larger than the 0.5th power mean of the length of edges $L_{\mathbf{EG}}$ and $L_{\mathbf{ED}}$). This procedure is repeated until a predefined size is achieved. The final individuals in the archive are **A**, **G**, and **H**. Clearly, by continuous

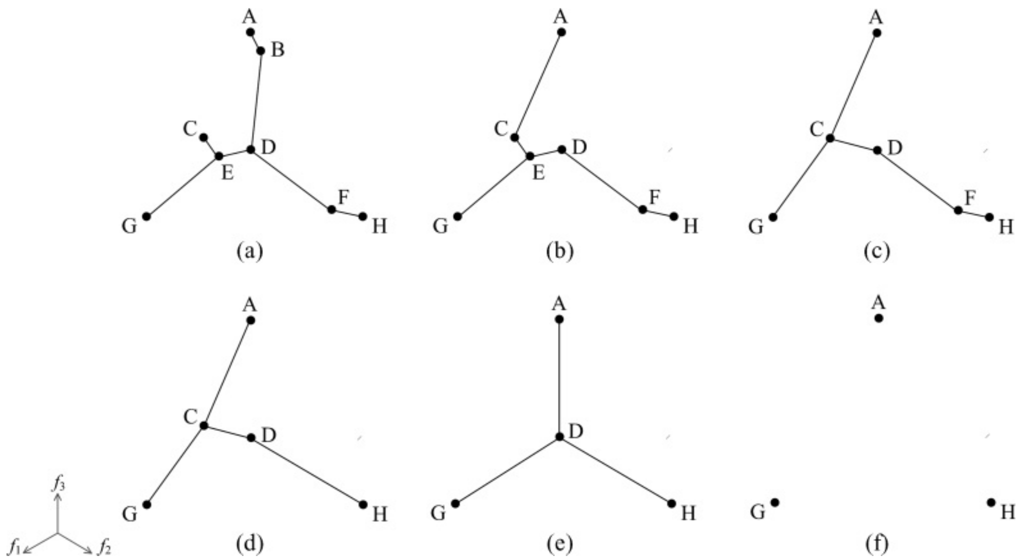

Figure 5: An example of the archive truncation process on a tri-objective non-dominated set, where the archive size is set to 3. (a) Original non-dominated set, (b) **B**-eliminated set, (c) **E**-eliminated set, (d) **F**-eliminated set, (e) **C**-eliminated set, and (f) **D**-eliminated set (i.e., the final individuals in the archive).

truncation in the archive, the two properties of distribution can be reasonably tuned, and a well-extended and uniformly-distributed non-dominated front will be obtained. More specifically, from the algorithm and illustration of archive truncation, we can draw some in-depth observations as follows.

1. Duplicate individuals, if they exist, will first be eliminated. This is because the edge weight between them is equal to zero in an EMST, and thus they would be selected to become the candidates first.

2. The comparison of degree information in the truncation strategy can be considered as a reasonable integration of the two properties of distribution, since it not only reflects the density of individuals but partly implies their position in the population. On the one hand, an individual of degree one, in general, means that it has a loose relationship with the surrounding individuals according to the property of EMSTs. Thus, its crowding extent is generally lower than that of the other individual sharing an edge. Obviously, preserving these individuals may be beneficial to the uniformity of distribution, in comparison with preserving their corresponding opponent. On the other hand, the boundary solutions (defined in Section 2) have a high likelihood of being preserved according to the degree comparison scheme. This is because they are located in the outer part of the population, and not all the orientations around them are with individuals, that is, only part of orientations may affect their degree. Therefore, for them, the probability of the degree equal to one is higher than that for the inner individuals. Figure 6 makes a statistical comparison between the boundary solutions and non-boundary solutions regarding the case that their degree is equal to one, considering 100 randomly generated non-dominated vectors (solutions) in the multidimensional space. It is clear that the probability (>70%) of the case that

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

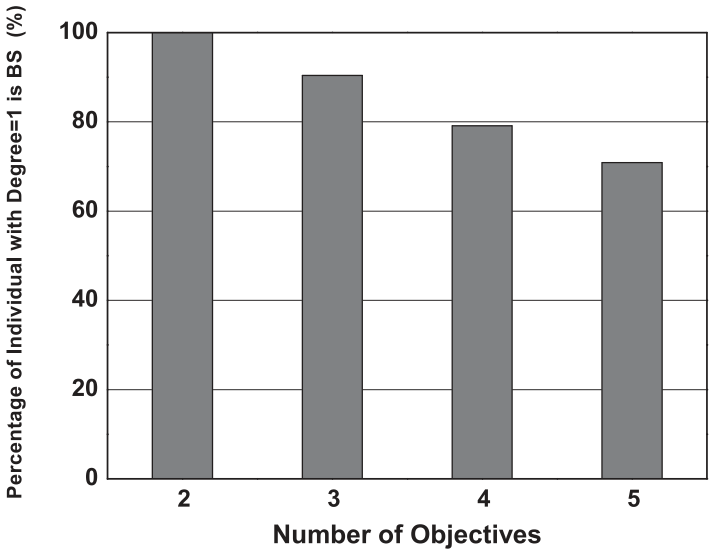

Figure 6: The percentage of the case that an individual of degree one is the boundary solution (BS), where the total case satisfies that, for a pair of individuals (i.e., two individuals sharing an edge in the EMST), one and only one belongs to BS, and one of them has the degree equal to one. The EMST is constructed by 100 non-dominated vectors which are randomly generated in the k-dimensional unit hypercube $[0, 1]^k$, where $k = 2, 3, 4, 5$.

the individual of degree one belongs to the boundary solutions is significantly larger than the probability (<30%) of the case that it belongs to the non-boundary solutions, especially in a lower dimension space. It is interesting to note that the probability reaches 100% in the two-dimensional space. This is because the EMST generated by two-dimensional non-dominated solutions is linear, and thus only two boundary solutions whose degree is equal to one exist.

3. When the degree of both candidates is larger than one, the modified ETCD, which takes into account their non-sharing edges, is introduced to determine their survival. In other words, we estimate the density of the two candidates by considering all individuals, except the closest one, connecting the candidates. This modification seems to be reasonable. In fact, there is always one candidate to be eliminated no matter how close the two candidates are, that is, the edge formed by them will not appear in the next round of truncation. Therefore, considering the effects of this edge is meaningless and may even lead to some erroneous judgments on their distributions. As the edge $L_{\mathbf{CD}}$ in Figure 5(d), the original ETCD of individual **C** is larger than that of individual **D**, and thus **D** will be eliminated. Obviously, this operation decreases the uniformity of solutions in the archive, compared to the result in Figure 5(e) obtained by the modified ETCD.

4 Experimental Design

This section is devoted to designing an experiment scheme for performance validation of ETEA. First, we briefly introduce the set of MOPs which will be used as the

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

benchmark for this experiment. Then, two popular metrics are described to give an appropriate performance evaluation for algorithms. Finally, a general experimental setting is presented for the comparison between ETEA and the other six EMO algorithms.

4.1 Test Problems

In this section, we describe different sets of test problems according to the number of objectives. These problems have been commonly used in the literature.

For the bi-objective problem set, we firstly choose problems from Van Veldhuizen's studies (Van Veldhuizen, 1999), including Schaffer1, Schaffer2, Fonseca, Kursawe, and Poloni. Then, the ZDT problem family, including ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 (Zitzler et al., 2000), is considered. Finally, the WFG problem family (WFG1 to WFG9) (Huband et al., 2006), based on variable linkages, is included. For the tri-objective problem set, three Viennet problems (Viennet1, Viennet2, and Viennet3; Van Veldhuizen, 1999) and the DTLZ problem family (DTLZ1 to DTLZ7; Deb, Thiele, et al., 2005) are chosen. Moreover, three recent tri-objective problems (called the UF problems; Zhang et al., 2009) which emphasize the complexity of the shapes of the Pareto set are taken into account as well. All the problems have been configured as in the original papers where they were described.

4.2 Performance Metrics

To compare the performance of the selected algorithms, we introduce two widely-used quality metrics, hypervolume (HV; Zitzler and Thiele, 1999) and inverted generational distance (IGD; Zhang et al., 2008), which can give a comprehensive assessment in terms of convergence, uniformity, and spread. The HV metric is a very popular quality metric due to its compliance with the Pareto dominance relation (see Zitzler et al., 2003). HV calculates the volume of the objective space between the obtained solution set and a reference point, and a larger value is preferable. On the other hand, IGD measures the average distance from the points in the Pareto front to their closest solution in the obtained set. A low IGD value indicates that the obtained solution set is close to the Pareto front and also has good distribution uniformity and range.

The main difference between IGD and HV is that, for the former, the Pareto front of problems must be known, and yet for the latter, a reference point that may bring about some effects on the performance judgment has to be chosen appropriately. In addition, the preference between uniformity and spread for the two metrics is also distinct. The IGD metric, which is based on uniformly-distributed points along the whole Pareto front, prefers the uniformity of the obtained solution set; while the HV metric, with significant contributions from the boundary solutions, has a bias toward the extent of the set.

4.3 General Experimental Setting

In order to validate the performance of ETEA, we compare it with six EMO algorithms: NSGA-II (Deb et al., 2002), SPEA2 (Zitzler et al., 2002), IBEA (Zitzler and Künzli, 2004), ϵ-MOEA (Deb, Mohan, et al., 2005), TDEA (Karahan and Köksalan, 2010), and MST-MOEA (i.e., the predecessor of ETEA; Li et al., 2008). NSGA-II[1] is one of the most popular EMO algorithms. The main characteristic of NSGA-II is its fast non-dominated sorting and crowding distance-based density estimation. SPEA2[2] is also a prevalent

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

[2]The C code of SPEA2 is available at *http://www.tik.ee.ethz.ch/pisa*

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

EMO algorithm, which borrows a so-called fitness strength value and the kth nearest neighbor to select individuals into the next population. In recent years, some indicator-based EMO algorithms have also found to be competitive in balancing convergence and diversity. Here, we select a representative indicator-based algorithm IBEA to make a comparative study. IBEA[3] aims to integrate the preference information of the decision-maker into multi-objective search. The main idea is to define the optimization goal in terms of a binary performance measure and then to directly use this measure in the mating and environmental selection processes. ϵ-MOEA[4] is a steady-state algorithm that typically creates only one new member that is tested to enter the population at each step of the algorithm (see Kumar and Rockett, 2002; Igel, Suttorp, et al., 2007; Durillo et al., 2009). ϵ-MOEA uses a grid-based strategy and divides the objective space into hyperboxes by the size of ϵ. Each hyperbox can contain at most a single individual, thus preventing crowding. However, due to the feature of ϵ-dominance, the boundary solutions may be lost in the evolutionary process (Hernández-Díaz et al., 2007; Karahan and Köksalan, 2010). Similar to ϵ-MOEA, TDEA[5] is also a grid-based steady-state algorithm. It defines a territory τ around an individual to maintain diversity. Its main difference against ϵ-MOEA is that the hyperboxes of TDEA are based on individuals rather than independent of them. The comparative study in Karahan and Köksalan (2010) shows its competitiveness in comparison with some state-of-the-art EMO algorithms. MST-MOEA is the first EMO algorithm that is designed based on the EMST. Although both MST-MOEA and ETEA algorithms employ the properties of EMSTs to enhance the performance of algorithms, they are of great difference in fitness assignment, environmental selection, and archive truncation. In the following, the experimental setting for the comparative study of these algorithms is listed.

- **Parameter Setting for Crossover and Mutation.** All selected EMO algorithms are given real-valued decision variables. Two widely-used crossover and mutation operators, simulated binary crossover (SBX) and polynomial mutation (Deb, 2001), are chosen. Following the practice in Deb et al. (2002), the distribution indexes in both SBX and the polynomial mutation are set to 20. A crossover probability $p_c = 1.0$ and a mutation probability $p_m = 1/n$ (where n is the number of decision variables) are used according to Deb (2001).

- **Population and Archive Size.** Like most of the studies of EMO algorithms, for generational algorithms the population size is set to 100, and the archive is also maintained at the same size if it exists (Coello et al., 2007). For steady-state algorithms, the regular population size is set to 100 according to Deb, Mohan, et al. (2005).

- **Number of Runs and Stopping Condition.** We independently run each algorithm 50 times for each test problem. The termination criterion of the algorithms is a predefined number of evaluations. Here, we set the evaluation number to different values for problems with different numbers of objectives, since the difficulty of problems generally increases with the number of objectives (Brockhoff et al., 2009; Schütze et al., 2011). Similar to the experimental

[3]The C code of IBEA is available at *http://www.tik.ee.thz.ch/pisa*

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

[5]The C code of TDEA was written by us.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 1: Parameter settings of ϵ-MOEA and TDEA.

	SCH1	SCH2	FON	KUR	POL	ZDT1	ZDT2	ZDT3
ϵ	0.0200	0.0180	0.0028	0.0350	0.0400	0.0076	0.0076	0.0030
τ	0.0110	0.0075	0.0130	0.0080	0.0080	0.0090	0.0090	0.0070
	ZDT4	ZDT6	WFG1	WFG2	WFG3	WFG4	WFG5	WFG6
ϵ	0.0075	0.0065	0.0070	0.0040	0.0200	0.0160	0.0160	0.0160
τ	0.0075	0.0060	0.0030	0.0070	0.0076	0.0100	0.0100	0.0100
	WFG7	WFG8	WFG9	VNT1	VNT2	VNT3	DTLZ1	DTLZ2
ϵ	0.0160	0.0110	0.0160	0.1000	0.0070	0.0110	0.0340	0.0630
τ	0.0100	0.0070	0.0100	0.0800	0.0260	0.0200	0.0600	0.1050
	DTLZ3	DTLZ4	DTLZ5	DTLZ6	DTLZ7	UF8	UF9	UF10
ϵ	0.0630	0.0150	0.0050	0.0300	0.0500	0.0150	0.0200	0.0050
τ	0.0200	0.0400	0.0110	0.0250	0.0600	0.0850	0.0700	0.0100

studies in Deb, Mohan, et al. (2005) and Beume et al. (2007), the algorithms are assigned a larger number of evaluations for tri-objective problems than for bi-objective ones, that is, 30,000 against 25,000.

- **Parameter Settings in IBEA, ϵ-MOEA, and TDEA.** In IBEA, the parameter κ is set to 0.05 as recommended in Zitzler and Künzli (2004). ϵ-MOEA and TDEA require the user to set the size of hyperboxes in grid (i.e., ϵ and τ). In order to guarantee a fair comparison, we set them so that the archive of the two algorithms is approximately of the same size as that of the other algorithms (given in Table 1).

- **Reference Point Setting in HV.** In the calculation of the HV metric for a solution set, choosing a reference point that is slightly larger than the worst value of each objective on the Pareto front is found to be suitable, since the effects of convergence and diversity of the set can be well balanced (Knowles, 2006; Auger et al., 2009). Here, as suggested in Kukkonen and Deb (2006), we select the integer point slightly larger than the worst value of each objective on the Pareto front of a problem as its reference point. As a consequence, the reference points for SCH1, SCH2, FON, KUR, and POL is (5, 5), (2, 17), (2, 2), (−14, 1), and (0, 1), respectively. The reference points used in all the ZDT and WFG problems is (2, 2) and (3, 5), respectively, and for VNT1, VNT2, and VNT3 is (5, 6, 5), (5, −16, −12), and (9, 18, 1), respectively. The reference point for the tri-objective DTLZ and UF problems is (2, 2, 2), except (1, 1, 1) for DTLZ1 and (2, 2, 7) for DTLZ7. Note that the solutions that do not dominate the reference point are discarded in the HV calculation (i.e., the solutions that are worse than the reference point in at least one objective will contribute zero to HV).

- **Substitution of the Pareto Front for IGD.** For the IGD metric, it is necessary to know the Pareto front of test problems. In most of the test problems used in this study, their Pareto fronts are known (families ZDT, DTLZ, WFG, and UF). For them we select 10,000 evenly-distributed points along the Pareto front as its substitution in the calculation of IGD since they can accurately represent the true Pareto front (Sen and Yang, 1998). For other test problems, the substitution of their Pareto fronts is available at the website http://www.cs.cinvestav.mx/emoobook/.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

- **Operating Environment.** The hardware used in the comparison experiments is a PC with 2.8 GHz Pentium 4 CPU with a memory of 1 GB, and the operating system is Windows XP. The code of ETEA and MST-MOEA is written in C.

5 Results and Discussion

This section validates the performance of ETEA according to the experimental design in the previous section. Firstly, we evaluate the proposed algorithm and compare it with five state-of-the-art EMO algorithms: NSGA-II, SPEA2, IBEA, ϵ-MOEA, and TDEA. Secondly, we analyze the time complexity of the proposed algorithm and show the computational cost of all the considered algorithms. Then, a comparative study between ETEA and its predecessor (MST-MOEA) is presented. Finally, we investigate the different parts of the proposed algorithm and identify their contribution to the performance of the algorithm.

5.1 Performance Comparison

In order to systematically present the results, the test problems have been grouped into two categories according to the number of their objectives. For each problem, we executed 50 independent runs. The values included in the tables of results are mean and standard deviation. The best mean for each problem has a gray background, as shown in Table 2. In addition, a t-test at a .05 significance level has been used to compare ETEA with its competitors. Symbols † and ‡ indicate that the p value of 98 DOF is significant at a .05 level by a two-tailed t-test. The symbol † indicates that ETEA is better than its competitor, and ‡ means the opposite.

Tables 2 and 3 show the results of the bi-objective problems in terms of HV and IGD, respectively. It is clear that ETEA performs significantly better than the other five EMO algorithms. For HV, the proposed algorithm obtains the best value in 13 out of the 19 test problems, and IBEA, NSGA-II, and SPEA2 perform the best in 3, 2, and 1 out of all the problems, respectively. Moreover, for the majority of the problems where the proposed algorithm outperforms the other algorithms, the results have statistical significance (12, 10, 13, 17, and 18 out of all the 19 problems for NSGA-II, SPEA2, IBEA, ϵ-MOEA, and TDEA, respectively). To graphically illustrate the work of these algorithms, we show typical distributions of the final solutions obtained by the six algorithms on ZDT4 and WFG6 in Figures 7 and 8, respectively. Clearly, the solutions of ETEA are located uniformly along the whole Pareto front of the problems, which means that ETEA can provide a good trade-off among convergence, uniformity, and spread.

Similar to HV, the results of IGD in Table 3 show that the proposed algorithm has a clear advantage over the other five algorithms for the majority of the problems. It obtains the best value in 13 out of the 19 problems, and most of the differences of the results between ETEA and the other algorithms have statistical significance. Specifically, the number of the problems where ETEA outperforms NSGA-II, SPEA2, IBEA, ϵ-MOEA, and TDEA with statistical significance is 16, 12, 17, 15, and 14, respectively. Interestingly, these algorithms sometimes obtain different and contradictory comparison results regarding different quality metrics (i.e., HV and IGD), although both metrics involve comprehensive performance of convergence, uniformity, and spread; for example, for WFG3, ETEA performs the best in terms of HV, but obtains a worse IGD value than ϵ-MOEA, whereas for WFG5, ETEA performs worse than IBEA with regard to HV but obtains the best IGD value of all.

In order to investigate such a contradictory observation, we introduce three widely-used performance metrics to separately assess the convergence, uniformity, and spread

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 2: The HV comparison of the six EMO algorithms on bi-objective problems.

Problem	ETEA	NSGA-II	SPEA2	IBEA	ϵ-MOEA	TDEA
SCH1	2.2275e+1 (6.47e−4)	2.2271e+1 (1.57e−3) †	2.2274e+1 (7.38e−4)	2.2272e+1 (1.07e−3) †	2.2229e+1 (1.07e−3) †	2.2270e+1 (2.02e−3) †
SCH2	3.8259e+1 (2.12e−3)	3.8246e+1 (3.84e−3) †	3.8258e+1 (2.41e−3) †	3.7981e+1 (1.32e−1) †	3.8121e+1 (1.34e−3) †	3.8219e+1 (2.98e−2) †
FON	3.0621e+0 (1.80e−4)	3.0618e+0 (1.84e−4) †	3.0620e+0 (1.16e−4)	3.0608e+0 (1.52e−4) †	3.0595e+0 (6.72e−4) †	3.0553e+0 (5.03e−3) †
KUR	3.7072e+1 (1.05e−2)	3.7005e+1 (1.42e−2) †	3.7064e+1 (1.15e−2) †	3.6662e+1 (5.63e−2) †	3.7068e+1 (1.46e−2) †	3.7050e+1 (2.13e−2) †
POL	7.5316e+1 (4.33e−2)	7.5267e+1 (5.59e−2) †	7.5326e+1 (9.72e−2)	6.0192e+1 (1.20e+0) †	7.0977e+1 (7.32e−2) †	7.4259e+1 (6.56e−1) †
ZDT1	3.6601e+0 (3.92e−4)	3.6591e+0 (4.10e−4) †	3.6594e+0 (4.72e−4) †	3.6590e+0 (8.02e−4) †	3.6476e+0 (1.73e−3) †	3.6566e+0 (1.76e−3) †
ZDT2	3.3260e+0 (6.68e−4)	3.3250e+0 (5.79e−4) †	3.3248e+0 (8.92e−4) †	3.3239e+0 (2.75e−4) †	3.3230e+0 (1.60e−3) †	3.3191e+0 (3.05e−3) †
ZDT3	4.8131e+0 (4.47e−4)	4.8124e+0 (4.66e−4) †	4.8118e+0 (5.16e−4) †	4.8062e+0 (2.11e−4) †	4.8094e+0 (1.11e−3) †	4.8035e+0 (3.61e−1) †
ZDT4	3.6514e+0 (7.73e−3)	3.6506e+0 (7.75e−3) †	3.6500e+0 (8.68e−3) †	2.4820e+0 (2.09e−1) †	3.6350e+0 (2.16e−2) †	3.6307e+0 (4.11e−2) †
ZDT6	3.0242e+0 (2.58e−3)	3.0219e+0 (2.72e−3) †	3.0230e+0 (2.10e−3) †	3.0365e+0 (5.70e−4) ‡	3.0281e+0 (2.22e−3) ‡	3.0236e+0 (2.68e−3) †
WFG1	7.2435e+0 (1.26e+0)	7.6348e+0 (9.96e−1)	7.5547e+0 (9.22e−1)	7.0554e+0 (9.81e−1)	5.6509e+0 (7.41e−1) †	5.4301e+0 (8.11e−1) †
WFG2	1.1151e+1 (4.17e−1)	1.1001e+1 (4.16e−1)	1.0952e+1 (4.09e−1)	1.0947e+1 (4.08e−1)	1.0914e+1 (4.02e−1)	1.0858e+1 (3.83e−1)
WFG3	1.0944e+1 (5.11e−3)	1.0934e+1 (7.03e−3) †	1.0940e+1 (5.52e−3) †	1.0941e+1 (3.09e−3) †	1.0926e+1 (8.49e−3) †	1.0917e+1 (1.35e−2) †
WFG4	8.6679e+0 (7.34e−3)	8.6676e+0 (4.01e−3)	8.6674e+0 (4.93e−3)	8.6671e+0 (1.64e−3)	8.6574e+0 (1.16e−2) †	8.6507e+0 (1.50e−2) †
WFG5	8.1575e+0 (3.00e−2)	8.1586e+0 (3.45e−2)	8.1531e+0 (3.49e−2)	8.1953e+0 (4.80e−2) ‡	8.1283e+0 (2.12e−2) †	8.1219e+0 (2.74e−2) †
WFG6	8.5708e+0 (1.07e−1)	8.5381e+0 (1.46e−1)	8.5088e+0 (1.57e−1)	8.4984e+0 (1.97e−1) †	8.4775e+0 (1.71e−1) †	8.4326e+0 (2.24e−1) †
WFG7	8.6703e+0 (6.64e−3)	8.6701e+0 (3.02e−3)	8.6689e+0 (6.54e−3) †	8.6675e+0 (1.50e−3) †	8.6612e+0 (1.05e−2) †	8.6488e+0 (1.44e−2) †
WFG8	7.0008e+0 (3.61e−1)	7.1049e+0 (4.50e−1) ‡	6.9988e+0 (4.42e−1)	6.9244e+0 (4.03e−1) †	6.8374e+0 (3.25e−1) †	6.7856e+0 (2.60e−1) †
WFG9	8.4377e+0 (1.57e−2)	8.4327e+0 (1.71e−2) †	8.4328e+0 (1.48e−2) †	8.4435e+0 (2.12e−2) ‡	8.4143e+0 (2.29e−2) †	8.4065e+0 (2.27e−2) †

†The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is better than its competitor.
‡The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is worse than its competitor.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 3: IGD comparison of the six EMO algorithms on bi-objective problems.

Problem	ETEA	NSGA-II	SPEA2	IBEA	ϵ-MOEA	TDEA
SCH1	1.6600e−2 (9.31e−5)	1.8770e−2 (4.11e−4)†	1.6607e−2 (1.04e−4)†	1.9027e−2 (4.10e−4)†	5.5664e−2 (6.14e−4)†	1.7382e−2 (4.66e−4)†
SCH2	2.2344e−2 (4.41e−4)	2.3655e−2 (9.19e−4)†	2.2645e−2 (5.44e−4)†	1.2724e−1 (5.64e−2)†	2.3706e−2 (1.23e−5)†	2.2486e−2 (3.26e−4)†
FON	4.6455e−3 (7.14e−5)	5.5651e−3 (1.94e−4)†	4.6601e−3 (7.31e−5)†	2.2760e−2 (2.46e−3)†	1.6571e−2 (1.07e−4)†	4.7148e−3 (1.37e−4)†
KUR	3.3764e−2 (6.57e−4)	4.2330e−2 (2.03e−3)†	3.4165e−2 (8.00e−4)†	2.0370e−1 (2.30e−2)†	3.5053e−2 (1.92e−4)†	3.4003e−2 (1.10e−3)†
POL	5.3160e−2 (1.36e−3)	6.9675e−2 (4.57e−3)†	5.3148e−2 (1.17e−3)	4.6631e−1 (1.32e−1)	1.9646e−1 (1.18e−3)†	6.1638e−2 (2.38e−3)†
ZDT1	4.0241e−3 (6.94e−5)	4.8165e−3 (2.27e−4)†	4.1792e−3 (9.21e−5)†	4.1447e−3 (6.10e−5)†	4.2747e−3 (5.43e−5)†	4.2314e−3 (1.43e−4)†
ZDT2	4.0065e−3 (7.01e−5)	4.8254e−3 (1.63e−4)†	4.1685e−3 (1.06e−4)†	9.2889e−3 (4.18e−4)†	5.7034e−3 (1.75e−4)†	4.3642e−3 (1.59e−4)†
ZDT3	4.9152e−3 (1.07e−4)	5.6877e−3 (2.93e−3)†	5.5663e−3 (4.13e−3)†	3.1754e−2 (3.40e−3)†	8.3480e−3 (9.02e−3)†	5.1557e−3 (7.05e−3)†
ZDT4	6.0413e−3 (2.05e−3)	6.5646e−3 (1.70e−3)†	6.5017e−3 (2.09e−3)†	6.1194e−1 (1.14e−1)†	6.9474e−3 (3.77e−3)†	7.5864e−3 (1.10e−2)†
ZDT6	6.8528e−3 (6.51e−4)	7.6825e−3 (7.41e−4)†	7.2374e−3 (5.67e−4)†	5.5267e−3 (1.59e−4)‡	5.1994e−3 (3.01e−4)‡	6.2396e−3 (5.51e−4)‡
WFG1	7.2727e−1 (2.09e−1)	6.1325e−1 (1.71e−1)‡	6.6093e−1 (1.58e−1)‡	8.6697e−1 (1.51e−1)†	1.0243e+0 (1.35e−1)†	1.0668e+0 (1.68e−1)†
WFG2	1.2190e−2 (1.82e−3)	1.4041e−2 (1.76e−3)†	1.2986e−2 (1.81e−3)†	7.3349e−2 (1.00e−2)†	1.3516e−2 (2.72e−3)†	1.4253e−2 (2.66e−3)†
WFG3	1.2146e−2 (3.77e−4)	1.4915e−2 (8.48e−4)†	1.2383e−2 (3.61e−4)†	1.2924e−2 (2.03e−4)†	1.1827e−2 (2.94e−4)‡	1.1923e−2 (4.24e−4)‡
WFG4	1.2945e−2 (2.50e−4)	1.3439e−2 (7.46e−4)†	1.2913e−2 (3.72e−4)‡	1.8421e−2 (1.06e−3)†	1.0121e−2 (9.64e−5)‡	1.1390e−2 (3.76e−4)‡
WFG5	6.6740e−2 (2.14e−4)	6.7911e−2 (1.60e−3)†	6.6761e−2 (1.13e−3)	7.1200e−2 (2.96e−4)†	6.8338e−2 (3.63e−5)†	6.6783e−2 (9.40e−5)
WFG6	2.6514e−2 (1.44e−2)	3.0866e−2 (2.04e−2)	3.3049e−2 (2.28e−2)	4.1843e−2 (2.82e−2)†	3.8578e−2 (2.46e−2)†	4.1992e−2 (3.36e−2)†
WFG7	1.3016e−2 (2.72e−4)	1.6155e−2 (8.32e−4)†	1.3064e−2 (2.95e−4)	2.1030e−2 (9.61e−4)†	1.6374e−2 (1.20e−4)†	1.3889e−2 (4.07e−4)
WFG8	1.7005e−1 (3.28e−2)	1.6018e−1 (4.32e−2)	1.6954e−1 (3.99e−2)	1.9361e−1 (2.75e−2)†	1.7821e−1 (3.07e−2)	1.8732e−1 (2.12e−2)†
WFG9	1.3875e−2 (1.18e−3)	1.7041e−2 (1.68e−3)†	1.4064e−2 (1.06e−3)†	1.9743e−2 (1.65e−3)†	1.6912e−2 (1.95e−3)†	1.5129e−2 (1.52e−3)†

†The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is better than its competitor.
‡The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is worse than its competitor.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

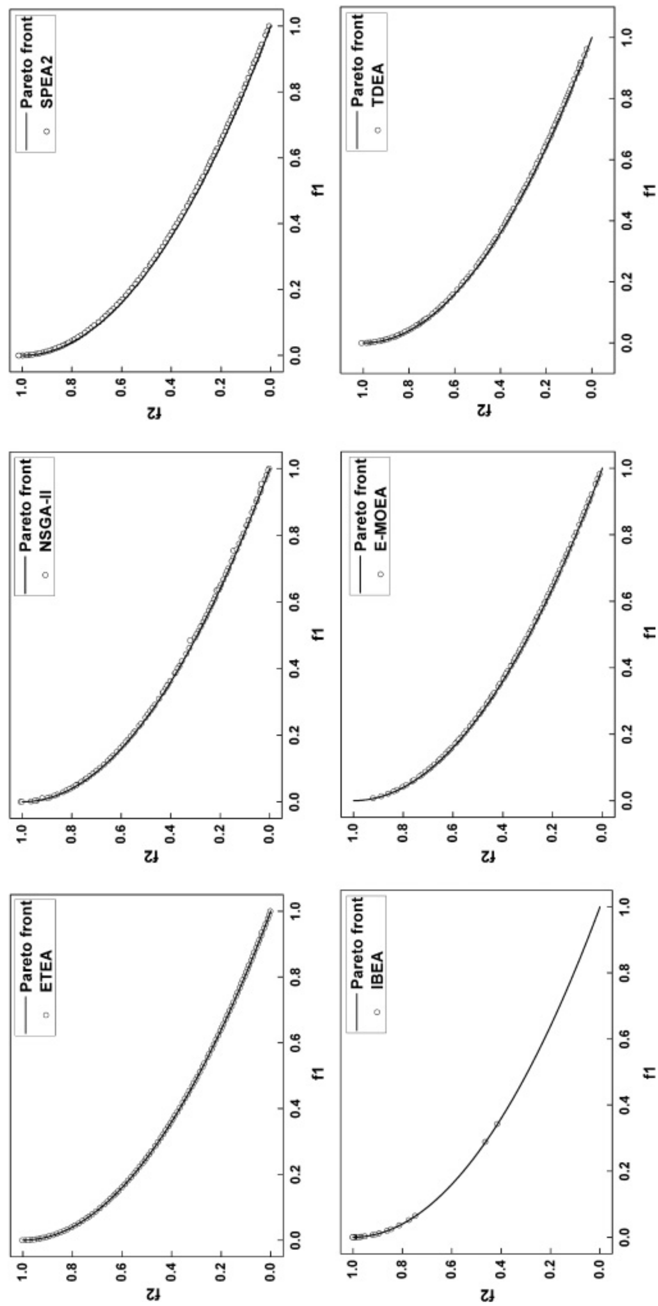

Figure 7: The final solutions obtained by the six algorithms on ZDT4.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

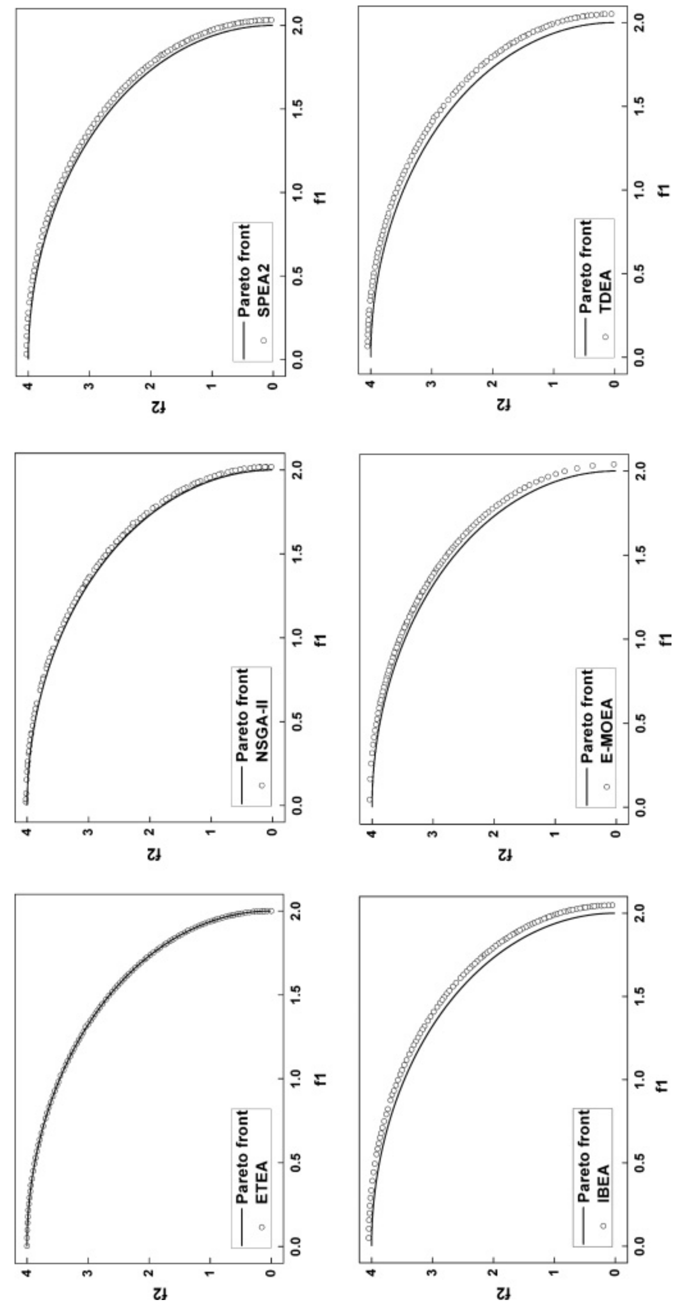

Figure 8: The final solutions obtained by the six algorithms on WFG6.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

of the solution sets. They are generational distance (GD; Van Veldhuizen and Lamont, 1998), spacing[6] (SP; Schott, 1995), and maximum spread[7] (MS; Zitzler et al., 2000). The GD metric evaluates the convergence of a solution set by measuring the average distance from the solutions in the set to their closest point in the Pareto front; SP evaluates the uniformity of a solution set by calculating the standard deviation of the distance from each solution to its closest neighbor in the set; and MS evaluates the spread of a solution set by measuring the length of the diagonal of a minimal hyperbox that encloses the set. For the former two metrics, a smaller value is preferable, and as to the last metric, a larger value is better. More details of these metrics can be found in Van Veldhuizen and Lamont (1998), Schott (1995), and Zitzler et al. (2000).

Here, the WFG problem family is selected for investigation since the contradictory phenomenon on it appears to be the most obvious. Table 4 gives the results of all the algorithms on the WFG problems in terms of GD, SP, and MS. Additionally, for a clearer comparison, the table shows the rank of the six algorithms for each problem according to their average value.

It can be seen from the table that in contrast to ETEA, NSGA-II, and SPEA2, the algorithms IBEA, ϵ-MOEA, and TDEA show clear differences among convergence, uniformity, and spread. IBEA performs the best in terms of MS but obtains the worst results with respect to GD and SP. ϵ-MOEA and TDEA perform well in terms of GD but have the worst values for the MS metric.

From the above observations, the contradictory results of the different algorithms on the comprehensive performance metrics HV and IGD can be reasonably explained, considering that HV prefers extensity and IGD has a bias toward distribution uniformity. IBEA, which directly selects individuals according to their fitness based on a binary performance measure, may fail to maintain the uniformity of a solution set, thus obtaining a relatively poor IGD result; ϵ-MOEA and TDEA, which lack extensity-preserving mechanisms, may lose the boundary solutions of the Pareto front, thus providing a relatively low HV value.

On the other hand, the results in Table 4 also show that the proposed algorithm ETEA is competitive for all the considered metrics. It takes the second, first, and third places in terms of GD, SP, and MS, respectively, regarding the sum rank on all WFG problems. This indicates that the solution set obtained by ETEA has a good balance among convergence, uniformity, and spread.

Tables 5 and 6 show the results for the tri-objective problems in terms of HV and IGD. The advantage of ETEA over the other algorithms on tri-objective problems seems to be not as clear as that for bi-objective problems. Nevertheless, ETEA performs better than the other five algorithms on more than half of all the problems. It is able to obtain the best values in 8 and 7 out of the 13 problems for HV and IGD, respectively, and with statistical significance in most of the cases. The number of the problems where ETEA outperforms NSGA-II, SPEA2, IBEA, ϵ-MOEA, and TDEA with statistical significance is 13, 10, 11, 8, and 8, respectively, for HV, and 13, 8, 10, 10, and 7, respectively, for IGD. Figure 9 gives a typical distribution of the final solutions obtained by the six algorithms on DTLZ1. For a better observation, the boundary of the Pareto front of the problem is also plotted in the figure.

[6]The spacing metric is also called SS or S in some literature (see Knowles and Corne, 2002; Kukkonen and Deb, 2006; Wei and Zhang, 2011).

[7]The maximum spread metric is called M_3^*, D, or FS in some literature (see Okabe et al., 2003; Kukkonen and Deb, 2006; Wei and Zhang, 2011).

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 4: Comparison of the GD, SP, and MS results of the six EMO algorithms on WFG problems. The number in the upper right quadrant associated with an algorithm indicates its rank among the six algorithms for a test problem. The value in the last row of each metric corresponds to the sum rank for all the problems.

Metric	problem	ETEA^{rank}	NSGA-II^{rank}	SPEA2^{rank}	IBEA^{rank}	ϵ-MOEA ^{rank}	TDEA^{rank}
GD	WFG1	5.263e−2 (1.5e−2)^{1}	5.349e−2 (1.7e−2)^{2}	6.778e−2 (1.8e−2)^{4}	6.076e−2 (1.9e−2)^{3}	7.104e−2 (1.6e−2)^{5}	9.505e−2 (2.0e−2)^{6}
	WFG2	8.723e−4 (8.0e−4)^{3}	1.027e−3 (8.8e−4)^{5}	8.442e−4 (7.8e−4)^{2}	1.577e−3 (1.6e−3)^{6}	9.646e−4 (9.6e−4)^{4}	6.492e−4 (5.3e−4)^{1}
	WFG3	5.167e−4 (7.3e−5)^{3}	6.261e−4 (6.7e−5)^{5}	5.388e−4 (5.7e−5)^{4}	6.963e−4 (1.3e−4)^{6}	4.528e−4 (5.5e−5)^{2}	3.519e−4 (9.7e−5)^{1}
	WFG4	1.470e−3 (6.3e−5)^{5}	1.404e−3 (1.5e−4)^{4}	1.529e−3 (8.0e−5)^{6}	9.975e−4 (1.5e−4)^{2}	9.955e−4 (2.0e−4)^{1}	1.301e−3 (1.2e−4)^{3}
	WFG5	6.280e−3 (3.1e−5)^{1}	6.440e−3 (3.8e−5)^{4}	6.296e−3 (2.6e−5)^{2}	6.963e−3 (3.6e−4)^{6}	6.714e−3 (4.5e−4)^{5}	6.316e−3 (7.5e−5)^{3}
	WFG6	1.830e−3 (2.5e−3)^{1}	1.989e−3 (1.7e−3)^{2}	2.966e−3 (2.5e−3)^{4}	3.221e−3 (3.7e−3)^{6}	2.570e−3 (2.0e−3)^{3}	3.168e−3 (2.6e−3)^{5}
	WFG7	7.394e−4 (3.9e−5)^{3}	8.650e−4 (7.1e−5)^{5}	7.564e−4 (4.4e−5)^{4}	9.937e−4 (3.3e−4)^{6}	5.347e−4 (5.7e−5)^{1}	7.048e−4 (5.4e−5)^{2}
	WFG8	2.772e−2 (5.6e−3)^{5}	2.566e−2 (6.9e−3)^{2}	2.970e−2 (7.6e−3)^{6}	2.714e−2 (5.0e−3)^{3}	2.394e−2 (3.7e−3)^{1}	2.769e−2 (7.8e−3)^{4}
	WFG9	7.261e−4 (2.9e−4)^{3}	1.020e−3 (2.5e−4)^{5}	7.475e−4 (1.4e−4)^{4}	1.258e−3 (4.8e−4)^{6}	5.997e−4 (1.6e−4)^{1}	6.635e−4 (2.0e−4)^{2}
	Sum rank	25	34	36	44	23	27
SP	WFG1	5.001e−3 (2.5e−3)^{1}	1.289e−2 (6.1e−3)^{3}	1.441e−2 (2.9e−2)^{5}	1.826e−2 (1.2e−2)^{6}	1.382e−2 (2.7e−2)^{4}	9.298e−3 (3.9e−3)^{2}
	WFG2	5.612e−3 (9.3e−4)^{1}	1.403e−2 (1.8e−3)^{4}	5.952e−3 (8.7e−4)^{2}	1.977e−2 (2.0e−3)^{6}	1.754e−2 (5.4e−4)^{5}	6.387e−3 (1.3e−3)^{3}
	WFG3	6.413e−3 (6.7e−4)^{3}	1.512e−2 (1.7e−3)^{6}	6.794e−3 (7.5e−4)^{4}	1.353e−2 (1.2e−3)^{5}	3.254e−3 (3.6e−4)^{1}	6.238e−3 (9.2e−4)^{2}
	WFG4	7.166e−3 (9.2e−4)^{1}	1.826e−2 (1.7e−3)^{4}	7.689e−3 (7.5e−4)^{2}	7.484e−2 (2.1e−2)^{6}	3.450e−2 (2.3e−3)^{5}	1.295e−2 (1.4e−3)^{3}
	WFG5	7.364e−3 (6.5e−4)^{1}	1.854e−2 (2.4e−3)^{4}	7.395e−3 (8.1e−4)^{2}	4.205e−2 (2.2e−2)^{6}	3.564e−2 (6.2e−4)^{5}	1.182e−2 (7.9e−4)^{3}
	WFG6	7.238e−3 (8.5e−4)^{1}	1.934e−2 (1.6e−3)^{4}	7.482e−3 (9.4e−4)^{2}	4.519e−2 (2.5e−2)^{5}	4.613e−2 (7.8e−4)^{6}	1.277e−2 (1.2e−3)^{3}
	WFG7	7.408e−3 (8.2e−4)^{1}	1.955e−2 (1.8e−3)^{4}	7.494e−3 (7.2e−4)^{2}	7.001e−2 (2.1e−2)^{6}	3.192e−2 (1.2e−3)^{5}	1.222e−2 (9.8e−4)^{3}
	WFG8	1.422e−2 (8.0e−3)^{2}	1.891e−2 (6.0e−3)^{3}	1.205e−2 (4.7e−3)^{1}	6.556e−2 (5.5e−2)^{6}	4.531e−2 (2.9e−2)^{5}	2.492e−2 (1.4e−2)^{4}
	WFG9	7.020e−3 (8.5e−4)^{1}	1.782e−2 (1.9e−3)^{4}	7.615e−3 (8.5e−4)^{2}	2.831e−2 (7.5e−3)^{5}	3.468e−2 (4.7e−3)^{6}	1.210e−2 (1.3e−3)^{3}
	Sum rank	12	36	22	51	42	26
MS	WFG1	2.822e+0 (7.8e−1)^{4}	3.021e+0 (7.6e−1)^{1}	2.962e+0 (6.3e−1)^{2}	2.833e+0 (6.3e−1)^{3}	2.113e+0 (6.3e−1)^{5}	1.931e+0 (6.8e−1)^{6}
	WFG2	3.926e+0 (4.9e−1)^{1}	3.920e+0 (5.0e−1)^{2}	3.860e+0 (4.8e−1)^{3}	3.846e+0 (4.8e−1)^{4}	3.797e+0 (4.9e−1)^{5}	3.665e+0 (3.9e−1)^{6}
	WFG3	4.469e+0 (3.2e−3)^{4}	4.470e+0 (5.0e−4)^{2}	4.470e+0 (2.9e−3)^{2}	4.472e+0 (6.8e−5)^{1}	4.436e+0 (4.3e−3)^{5}	4.424e+0 (1.8e−3)^{6}
	WFG4	4.471e+0 (2.3e−3)^{1}	4.470e+0 (1.1e−3)^{2}	4.469e+0 (2.4e−3)^{4}	4.470e+0 (2.6e−4)^{2}	4.459e+0 (9.4e−3)^{5}	4.456e+0 (8.9e−3)^{6}
	WFG5	4.409e+0 (1.9e−2)^{3}	4.410e+0 (1.9e−2)^{2}	4.409e+0 (1.8e−2)^{3}	4.431e+0 (2.1e−2)^{1}	4.386e+0 (1.6e−2)^{6}	4.387e+0 (2.2e−2)^{5}
	WFG6	4.472e+0 (1.3e−3)^{1}	4.471e+0 (1.2e−3)^{3}	4.471e+0 (1.8e−3)^{3}	4.472e+0 (4.1e−3)^{1}	4.463e+0 (7.0e−3)^{5}	4.455e+0 (6.8e−3)^{6}
	WFG7	4.471e+0 (3.5e−3)^{2}	4.471e+0 (5.8e−4)^{2}	4.470e+0 (1.5e−3)^{4}	4.472e+0 (1.5e−4)^{1}	4.460e+0 (7.5e−3)^{5}	4.450e+0 (9.7e−3)^{6}
	WFG8	4.394e+0 (6.3e−2)^{3}	4.416e+0 (6.4e−2)^{1}	4.384e+0 (4.6e−2)^{4}	4.402e+0 (6.5e−2)^{2}	4.362e+0 (7.1e−2)^{5}	4.339e+0 (9.2e−2)^{6}
	WFG9	4.327e+0 (1.7e−2)^{2}	4.326e+0 (2.0e−2)^{4}	4.327e+0 (1.8e−2)^{2}	4.330e+0 (1.3e−2)^{1}	4.309e+0 (9.6e−3)^{5}	4.305e+0 (1.8e−2)^{6}
	Sum rank	21	19	27	16	46	53

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 5: HV comparison of the six EMO algorithms on tri-objective problems.

Problem	ETEA	NSGA-II	SPEA2	IBEA	ϵ-MOEA	TDEA
VNT1	6.1582e+1 (4.15e−2)	6.1184e+1 (1.07e−1)†	6.1577e+1 (3.35e−2)	5.9847e+1 (7.20e−2)†	6.0071e+1 (3.49e−2)†	6.1522e+1 (7.36e−2)
VNT2	1.9146e+0 (3.90e−4)	1.9094e+0 (1.99e−3)†	1.9145e+0 (5.62e−4)	1.8743e+0 (1.60e−2)†	1.9069e+0 (1.84e−4)†	1.9159e+0 (3.16e−4)‡
VNT3	2.8303e+1 (1.09e−2)	2.8271e+1 (4.89e−3)†	2.8300e+1 (1.01e−2)‡	2.8124e+1 (2.57e−2)†	2.8365e+1 (7.09e−3)†	2.8353e+1 (3.47e−3)‡
DTLZ1	9.7286e−1 (3.30e−4)	9.6730e−1 (5.35e−4)†	9.7212e−1 (1.08e−3)†	8.9767e−1 (1.01e−2)†	9.5612e−1 (1.26e−2)†	9.7072e−1 (5.13e−4)†
DTLZ2	7.3948e+0 (6.57e−3)	7.3512e+0 (1.98e−2)†	7.3912e+0 (6.93e−3)†	5.7020e+0 (2.92e+0)†	7.3890e+0 (9.06e−3)†	7.4050e+0 (1.18e−2)‡
DTLZ3	2.3436e+0 (2.75e+0)	6.9185e−1 (2.33e+0)†	6.3694e−1 (1.79e+0)†	6.2280e+0 (2.11e+0)‡	6.3262e+0 (2.09e+0)‡	9.8252e−1 (2.17e+0)†
DTLZ4	7.2158e+0 (3.19e−1)	6.8874e+0 (5.94e−1)†	6.9116e+0 (5.53e−1)†	6.4116e+0 (1.19e+0)†	7.0132e+0 (4.04e−1)†	6.9417e+0 (4.54e−1)†
DTLZ5	6.1015e+0 (6.15e−4)	6.0993e+0 (6.83e−4)†	6.1011e+0 (7.26e−4)†	6.0884e+0 (1.06e−3)†	6.1000e+0 (2.11e−3)†	6.1013e+0 (1.23e−3)†
DTLZ6	4.6522e+0 (1.66e−1)	4.0978e+0 (2.60e−1)†	4.0443e+0 (2.66e−1)†	5.9742e+0 (9.68e−2)‡	5.2916e+0 (1.27e−1)‡	5.2551e+0 (1.12e−1)‡
DTLZ7	1.3400e+1 (2.60e−2)	1.3258e+1 (5.74e−2)†	1.3331e+1 (2.06e−1)†	1.0257e+1 (2.73e+0)†	1.3120e+1 (2.04e−2)†	1.3275e+1 (4.80e−2)†
UF8	6.9641e+0 (3.83e−1)	6.4670e+0 (2.62e−1)†	6.7052e+0 (3.58e−1)†	6.6493e+0 (3.36e−1)†	6.8654e+0 (3.09e−1)†	6.6603e+0 (2.44e−1)†
UF9	7.1990e+0 (3.41e−1)	6.0702e+0 (6.33e−1)†	6.8345e+0 (3.97e−1)†	6.3068e+0 (8.74e−2)†	7.1966e+0 (2.46e−1)	7.0881e+0 (2.55e−1)†
UF10	5.2946e+0 (7.31e−1)	3.7563e+0 (1.56e+0)†	5.2645e+0 (6.98e−1)	4.8410e+0 (8.12e−1)†	5.0434e+0 (9.63e−1)	3.8190e+0 (8.95e−1)†

†The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is better than its competitor.
‡The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is worse than its competitor.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 6: IGD comparison of the six EMO algorithms on tri-objective problems.

Problem	ETEA	NSGA-II	SPEA2	IBEA	ϵ-MOEA	TDEA
VNT1	1.2664e−1 (2.38e−3)	1.5865e−1 (7.63e−3)†	1.2714e−1 (2.44e−3)	1.6788e−1 (2.66e−2)†	1.4595e−1 (9.38e−4)†	1.2965e−1 (2.89e−3)†
VNT2	1.2305e−2 (2.54e−4)	2.3127e−2 (2.22e−3)†	1.2311e−2 (3.39e−4)	4.4864e−2 (9.86e−3)†	1.6541e−2 (3.20e−4)†	1.1023e−2 (2.40e−4)‡
VNT3	3.2065e−2 (1.09e−3)	4.9850e−2 (2.98e−3)†	3.2478e−2 (1.27e−3)†	2.5818e+0 (9.51e−2)†	2.9258e−1 (3.20e−1)†	3.8531e−2 (3.17e−3)†
DTLZ1	2.0657e−2 (5.21e−4)	3.3593e−2 (3.12e−2)†	2.2111e−2 (2.38e−3)†	1.8333e−1 (1.51e−2)†	2.3695e−2 (2.26e−3)†	4.6974e−2 (9.50e−2)†
DTLZ2	5.4021e−2 (9.72e−4)	6.8904e−2 (3.10e−3)†	5.4357e−2 (1.30e−3)†	5.1119e−1 (5.53e−1)†	6.8102e−2 (9.32e−4)†	5.3236e−2 (1.58e−3)‡
DTLZ3	1.4106e+0 (1.10e+0)	2.7742e+0 (2.02e+0)†	1.7410e+0 (1.50e+0)	5.5839e−1 (2.46e−1)‡	3.5472e−1 (6.35e−1)‡	2.6762e+0 (2.14e+0)†
DTLZ4	1.5341e−1 (1.46e−1)	2.3645e−1 (1.66e−1)†	1.9619e−1 (1.73e−1)†	4.9543e−1 (3.69e−1)†	2.3741e−1 (1.62e−1)†	2.4994e−1 (1.74e−1)†
DTLZ5	4.2391e−3 (3.82e−4)	5.5382e−3 (3.31e−4)†	4.3894e−3 (4.12e−4)†	2.5685e−2 (1.32e−3)†	6.4302e−3 (1.50e−4)†	4.1865e−3 (3.08e−4)
DTLZ6	4.9256e−1 (4.74e−2)	6.5429e−1 (6.32e−2)†	6.5959e−1 (6.47e−2)†	6.2603e−2 (3.00e−2)‡	2.7887e−1 (4.60e−2)‡	2.8453e−1 (3.52e−2)‡
DTLZ7	6.2268e−2 (1.16e−3)	7.7085e−2 (4.22e−3)†	6.3198e−2 (3.67e−3)	4.4057e−1 (3.16e−1)†	7.3459e−2 (8.14e−2)†	6.4598e−2 (3.08e−2)
UF8	1.3654e−1 (4.47e−2)	2.3119e−1 (4.54e−2)†	1.4552e−1 (4.05e−2)†	3.3842e−1 (2.31e−2)†	2.4193e−1 (8.10e−2)†	1.4845e−1 (2.56e−2)†
UF9	1.6542e−1 (5.21e−2)	3.7618e−1 (9.30e−2)†	1.9727e−1 (6.24e−2)†	1.3921e−1 (4.76e−2)‡	1.2172e−1 (6.44e−2)‡	1.3805e−1 (4.50e−2)‡
UF10	3.1911e−1 (6.83e−2)	5.8537e−1 (2.82e−1)†	3.2101e−1 (5.78e−2)	5.5294e−1 (9.56e−2)†	4.3721e−1 (1.10e−1)†	4.2203e−1 (9.81e−2)†

†The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is better than its competitor.
‡The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is worse than its competitor.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

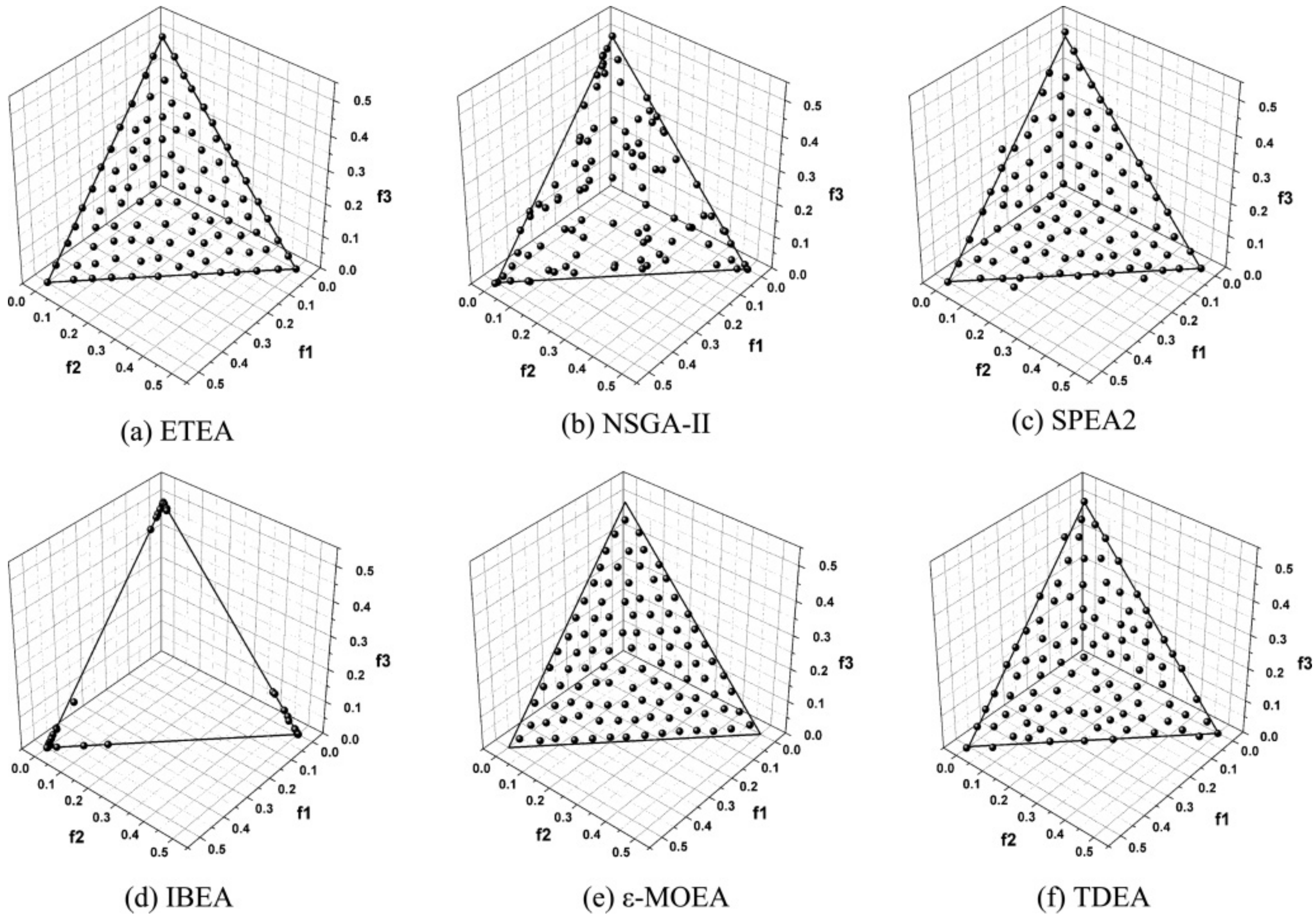

Figure 9: The final solutions obtained by the six algorithms on DTLZ1. The triangle corresponds to the boundary of the Pareto front of the problem.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Clearly, ETEA provides a good balance among convergence, uniformity, and spread. NSGA-II and IBEA perform poorly in maintaining diversity, and especially for IBEA, the solutions are grouped into three clusters which are around the extreme solutions of the Pareto front. The solutions obtained by ϵ-MOEA appear to be the most uniform, but they fail to expand to the boundary of the Pareto front. Considering SPEA2 and TDEA, although they perform well in terms of uniformity and spread, some of their solutions are located beyond the range of the Pareto front (i.e., fail to converge into the optimal front).

However, it should be pointed out that for some problems that are difficult to converge into the Pareto front, such as DTLZ3 and DTLZ6, the proposed algorithm performs worse than IBEA and ϵ-MOEA. This may be attributed to the environmental selection mechanism in ETEA that distinguishes solutions according to their distributions when they are non-dominated to each other, which can lead to the longtime existence of the dominance resistant solutions (i.e., the solutions with an extremely poor value in at least one of the objectives, but with near optimal values in the others; see Ikeda et al., 2001) in the population. These solutions will negatively affect the population searching toward the Pareto front to a large extent (Hanne, 2001; Purshouse and Fleming, 2007). This phenomenon happens more for tri-objective problems than for bi-objective problems.

Finally, note that for some of the complicated linkage problems (i.e., the WFG family and the UF family), all the considered EMO algorithms encounter great difficulty in reaching the Pareto front. One of the main reasons may be attributed to the fact that the SBX operator cannot work well on variable linkage problems (Deb et al., 2006). In spite of this, ETEA, compared to the other five algorithms, is able to obtain the best results in 8 and 7 out of the 12 WFG and UF problems for HV and IGD, respectively.

5.2 Computational Cost

As can be seen from the outline of ETEA in Algorithm 1, the computational cost of the algorithm is mainly determined by three steps: fitness assignment, fitness adjustment, and archive truncation. In the following, we will briefly analyze the computational cost of these steps.

- **Computational Cost of Fitness Assignment.** The computational cost of fitness assignment can be divided into two parts: the calculation of the distance count and the calculation of ETCD. Calculating the distance count needs to identify the nondominated front in the population of size N. This usually requires $O(MN^2)$ comparisons for all solutions in the population, where M is the number of objectives. In addition, for all dominated solutions, $O(MN^2)$ computations are required in order to find the nearest non-dominated solutions which dominate them. Finally, the distance comparison among the non-dominated solutions is introduced to define the distance count. This process also requires $O(MN^2)$ computations. Thus, the total complexity of calculating the distance count is $O(MN^2)$. Concerning the calculation of ETCD, the Euclidean distance between each pair of solutions is required to generate an EMST of the whole population. This requires $O(MN^2)$ computations. The time complexity is $O(N^2)$ for generating an EMST by the Prim algorithm and at the same time calculating ETCD of each individual. Thus, the whole process of the ETCD assignment requires $O(MN^2)$ computations, and further, the total computational cost of the fitness assignment is $O(MN^2)$.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 7: The average CPU time (s) used by each algorithm over 50 runs with/without the cost of 30,000 function evaluations on DTLZ2. The left column corresponds to the cost including the evaluations.

Algorithms	Bi-objective		Tri-objective	
ETEA	6.8284	6.8147	13.674	13.6432
NSGA-II	1.035	1.0203	1.271	1.2404
SPEA2	8.0651	8.0505	12.4923	12.4642
IBEA	3.1267	3.1129	3.683	3.6554
ϵ-MOEA	0.5294	0.5145	0.6859	0.6512
TDEA	0.7625	0.7458	0.9403	0.9085

- **Computational Cost of Fitness Adjustment.** The procedure of the fitness adjustment is implemented when the number of the non-dominated solutions is insufficient to fill the slots in the archive. The computational cost of this procedure is mainly determined by three functions: $Findout\_best$, $Findout\_neighbor$, and $Sort$ (cf. Algorithm 2). Clearly, $Findout\_best$ requires $O(N)$ comparisons. The time complexity of the functions $Findout\_neighbor$ and $Sort$ is $O(MN)$ and $O(N \log N)$, respectively. Thus, the total time complexity of this step is bounded by $O(MN^2)$ or $O(N^2 \log N)$, whichever is greater.

- **Computational Cost of Archive Truncation.** In contrast to fitness adjustment, archive truncation is activated when the number of the non-dominated solutions exceeds the upper bound of the archive. Similar to the calculation of ETCD, the Euclidean distance between each pair of solutions is required first, which needs $O(MN^2)$ computations. Since the process of archive truncation belongs to recurrence-mode (i.e., assessment has to be performed on the remaining solutions in each cycle of the truncation; see Khor et al., 2005), the time complexity is $O(N^3)$ (cf. Algorithm 3). Therefore, the total time complexity of this step is bounded by $O(MN^2)$ or $O(N^3)$, whichever is greater.

In conclusion, the overall complexity of ETEA is divided into two cases: if the fitness adjustment procedure is implemented, then the overall complexity is bounded by $O(MN^2)$ or $O(N^2 \log N)$, whichever is greater; if the archive truncation is online, then the overall complexity is $O(MN^2)$ or $O(N^3)$, whichever is greater. In most cases, $M < \log N$, and thus, the former is $O(N^2 \log N)$ and the latter is bounded by $O(N^3)$.

Table 7 shows the average computational cost of the six EMO algorithms on DTLZ2; additionally, the average CPU time without the cost of function evaluations is also contained in the table in order to understand the computational effort of the internal algorithm procedures. On each problem, the evaluation number is set to 30,000. The settings of other parameters are the same as those in the previous studies.

Clearly, the algorithms ϵ-MOEA, TDEA, NSGA-II, and IBEA take the first four places. The computational complexity of them is $O(MN)$, $O(MN)$, $O(MN^2)$, and $O(MN^2)$, respectively. ETEA and SPEA2 have similar computational costs, though ETEA performs better on bi-objective problems, while SPEA2 obtains lower values for problems with three objectives. This phenomenon may be due to the increasing proportion of non-dominated solutions in the population with the number of objectives. For the case in which the number of non-dominated solutions exceeds the archive size, the archive

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

truncation process will largely govern the computational cost of the algorithms, and the time complexity of ETEA is higher than that of SPEA2 ($O(N^3)$ against $O(N^2 \log N)$). For the case in which the number of non-dominated solutions is smaller than the archive size, although both algorithms have identical time complexity ($O(N^2 \log N)$), the density estimator in SPEA2 (i.e., the kth nearest neighbor) may take more time than the sorting function in ETEA which does not aim at all individuals in the population.

Although ETEA generally requires more time than the other EMO algorithms, it can be improved by several potential ways. The most direct way is to reduce the time complexity of constructing an EMST. Indeed, there currently exist some more efficient methods to speed up the construction of an EMST in comparison with the Prim algorithm, especially in the two-dimensional space (Agarwal et al., 1991; Eppstein, 1995; Czumaj et al., 2003, 2005; Rajasekaran, 2005). Another way of decreasing the computational cost of ETEA is to reduce the time complexity of the archive truncation process. Since an EMST for all remaining individuals has to be generated in each cycle of the truncation, the process requires $O(N^3)$ computations. Considering a smaller EMST, which is constructed by the individuals that used to be the neighbors of the eliminated individual, instead of the EMST of all remaining solutions, the time complexity of the truncation can be reduced to $O(MN^2)$. Previous effort in this direction has been reported in Li et al. (2009).

Finally, it is necessary to point out an interesting phenomenon in archive truncation of ETEA for bi-objective MOPs. In these problems, the EMST of non-dominated individuals degenerates into a linear structure. Figure 10 exemplifies the truncation process. Clearly, the degree of individuals in the EMST remains unchanged (i.e., the degree of boundary individuals is always one, and the degree of inner individuals is always two). Therefore, the boundary individuals in the truncation process will never be eliminated, and the inner ones will be tested only by the distance from their two neighbors. In this case, constructing an EMST seems to be unnecessary, and it can be replaced by renewing and comparing the neighbor information of individuals. This process only requires $O(MN)$ computations in each cycle. Therefore, the total time complexity of the archive truncation procedure is $O(MN^2)$. In fact, the truncation can still be speeded up, using some efficient data structure (e.g., heap) to orderly store the neighbor information of individuals. This is similar to Kukkonen and Deb (2006), where the population is rapidly maintained by heapsort. In this case, the time complexity of the archive truncation of ETEA for bi-objective MOPs can be reduced to $O(MN \log N)$.

Table 8 shows the computational costs of ETEA for all the bi-objective test problems after speeding up the archive truncation process according to the above method. The upper value in each cell corresponds to the cost of the original ETEA. It is clear from the table that for all the problems, the computational cost of the algorithm has a reduction of at least 12%. Interestingly, the effect of the acceleration of the archive truncation on different problems is apparently distinct. For some problems, such as SCH2, POL, and WFG5, the reduction rate of the runtime reaches 60%. This occurrence can be attributed to the difference of the numbers of the non-dominated solutions in environmental selection during the evolutionary process. If the number of the non-dominated solutions greatly exceeds the size of the archive set, the total time of ETEA will be significantly reduced by speeding up the archive truncation.

5.3 Comparison with Its Predecessor

MST-MOEA is the first attempt to capture and employ the properties of EMSTs for EMO. The comparison studies in Li et al. (2008) found that MST-MOEA is competitive

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

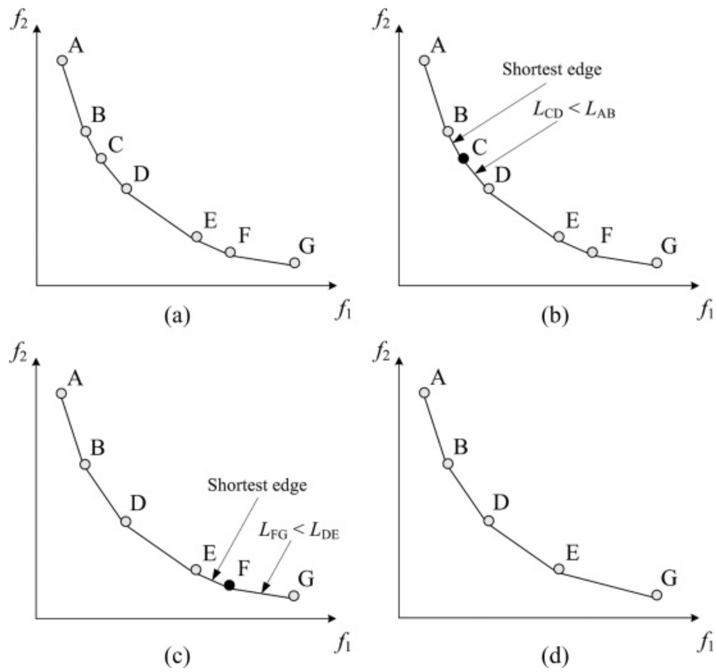

Figure 10: An example of the archive truncation in ETEA for a bi-objective nondominated set, where the archive size is 5. (a) The original non-dominated set; (b) **C** is eliminated since $L_{\mathbf{BC}}$ is the shortest and $L_{\mathbf{CD}} < L_{\mathbf{AB}}$; (c) **F** is eliminated since $L_{\mathbf{EF}}$ is the shortest and $L_{\mathbf{FG}} < L_{\mathbf{DE}}$; (d) The final archive set.

Table 8: Comparison of the computational costs of ETEA with and without the acceleration of the archive truncation for the bi-objective problems. The upper value in each cell corresponds to the cost without the acceleration.

Problem	Time	Problem	Time	Problem	Time	Problem	Time
SCH1	3.4886	ZDT1	6.3082	WFG1	9.3377	WFG6	8.3982
	1.9223		3.4838		6.8621		4.4378
SCH2	8.2630	ZDT2	5.8952	WFG2	7.7727	WFG7	11.629
	2.0765		4.0127		5.2836		6.2091
FON	5.9672	ZDT3	5.7923	WFG3	10.154	WFG8	7.9353
	2.2415		3.4681		5.1103		6.9505
KUR	4.4482	ZDT4	5.3873	WFG4	8.8742	WFG9	11.424
	2.1930		4.4862		4.7822		6.6900
POL	8.3503	ZDT6	5.6741	WFG5	12.165		
	2.0342		4.5270		4.7948		

with NSGA-II and SPEA2 in terms of convergence and diversity. In this section, we compare ETEA with this algorithm. Tables 9 and 10 show the HV and IGD results of the two algorithms. The better value for each problem is painted with a gray background.

It can be observed from the tables that ETEA has a clear advantage over its predecessor on most of the problems. For HV, ETEA obtains better values in 27 out of the 32

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 9: HV comparison results between MST-MOEA and ETEA.

	Algorithms			Algorithms	
Problem	MST-MOEA	ETEA	Problem	MST-MOEA	ETEA
SCH1	2.2274e+1 (7.20e−4)	2.2275e+1 (6.47e−4)	WFG7	8.6702e+0 (7.07e−3)	8.6703e+0 (6.64e−3)
SCH2	3.8258e+1 (2.16e−3)†	3.8259e+1 (2.12e−3)	WFG8	6.9259e+0 (3.86e−1)†	7.0008e+0 (3.61e−1)
FON	3.0620e+0 (1.16e−4)†	3.0621e+0 (1.80e−4)	WFG9	8.4416e+0 (1.80e−2)‡	8.4377e+0 (1.57e−2)
KUR	3.7071e+1 (1.18e−2)	3.7072e+1 (1.05e−2)	VNT1	6.1571e+1 (4.63e−2)†	6.1582e+1 (4.15e−2)
POL	7.5303e+1 (5.31e−2)†	7.5316e+1 (4.33e−2)	VNT2	1.9138e+0 (4.34e−4)†	1.9146e+0 (3.90e−4)
ZDT1	3.6598e+0 (5.42e−4)†	3.6601e+0 (3.92e−4)	VNT3	2.8305e+1 (9.25e−3)	2.8303e+1 (1.09e−2)
ZDT2	3.3238e+0 (1.18e−3)†	3.3260e+0 (6.68e−4)	DTLZ1	9.7263e−1 (6.53e−4)†	9.7286e−1 (3.30e−4)
ZDT3	4.8119e+0 (6.54e−4)†	4.8131e+0 (4.47e−4)	DTLZ2	7.3909e+0 (8.69e−3)†	7.3948e+0 (6.57e−3)
ZDT4	3.4475e+0 (1.60e−1)†	3.6514e+0 (7.73e−3)	DTLZ3	2.2463e+0 (3.04e+0)	2.3436e+0 (2.75e+0)
ZDT6	3.0162e+0 (5.49e−3)†	3.0242e+0 (2.58e−3)	DTLZ4	7.0395e+0 (4.46e−1)†	7.2158e+0 (3.19e−1)
WFG1	6.8129e+0 (9.98e−1)	7.2435e+0 (1.26e+0)	DTLZ5	6.1012e+0 (4.34e−4)†	6.1015e+0 (6.15e−4)
WFG2	1.0884e+1 (3.87e−1)†	1.1151e+1 (4.17e−1)	DTLZ6	4.6281e+0 (1.84e−1)	4.6522e+0 (1.66e−1)
WFG3	1.0941e+1 (6.23e−3)†	1.0944e+1 (5.11e−3)	DTLZ7	1.1827e+1 (1.99e−1)†	1.3400e+1 (2.60e−2)
WFG4	8.6680e+0 (7.47e−3)	8.6679e+0 (7.34e−3)	UF8	7.0190e+0 (3.58e−1)	6.9641e+0 (3.83e−1)
WFG5	8.1581e+0 (3.29e−2)	8.1575e+0 (3.00e−2)	UF9	6.8827e+0 (2.76e−1)†	7.1990e+0 (3.41e−1)
WFG6	8.4764e+0 (1.64e−1)†	8.5708e+0 (1.07e−1)	UF10	5.2137e+0 (6.82e−1)	5.2946e+0 (7.31e−1)

†The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is better than its competitor.
‡The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is worse than its competitor.

Table 10: IGD comparison results between MST-MOEA and ETEA.

	Algorithms			Algorithms	
Problem	MST-MOEA	ETEA	Problem	MST-MOEA	ETEA
SCH1	1.6604e−2 (9.86e−5)	1.6600e−2 (9.31e−5)	WFG7	1.3021e−2 (3.00e−4)	1.3016e−2 (2.72e−4)
SCH2	2.2449e−2 (5.33e−4)†	2.2344e−2 (4.41e−4)	WFG8	1.7341e−1 (3.40e−2)	1.7005e−1 (3.28e−2)
FON	4.6585e−3 (7.67e−5)†	4.6455e−3 (7.14e−5)	WFG9	1.3855e−2 (1.10e−3)	1.3875e−2 (1.18e−3)
KUR	3.3983e−2 (7.28e−4)†	3.3764e−2 (6.57e−4)	VNT1	1.3095e−1 (2.58e−3)†	1.2664e−1 (2.38e−3)
POL	5.3633e−2 (1.46e−3)†	5.3160e−2 (1.36e−3)	VNT2	1.2494e−2 (2.39e−4)†	1.2305e−2 (2.54e−4)
ZDT1	4.0634e−3 (8.10e−5)†	4.0241e−3 (6.94e−5)	VNT3	3.2256e−2 (1.13e−3)†	3.2065e−2 (1.09e−3)
ZDT2	4.0941e−3 (3.48e−4)†	4.0065e−3 (7.01e−5)	DTLZ1	2.1356e−2 (1.43e−3)†	2.0657e−2 (5.21e−4)
ZDT3	5.5663e−3 (4.13e−3)†	4.9152e−3 (1.07e−4)	DTLZ2	5.4460e−2 (1.11e−3)	5.4021e−2 (9.72e−4)
ZDT4	7.5463e−3 (6.87e−3)†	6.0413e−3 (2.05e−3)	DTLZ3	1.4639e+0 (1.21e+0)	1.4106e+0 (1.10e+0)
ZDT6	9.0741e−3 (1.53e−3)†	6.8528e−3 (6.51e−4)	DTLZ4	2.3510e−1 (1.75e−1)†	1.5341e−1 (1.46e−1)
WFG1	8.0317e−1 (1.58e−1)†	7.2727e−1 (2.09e−1)	DTLZ5	4.4181e−3 (1.17e−4)†	4.2391e−3 (3.82e−4)
WFG2	1.3277e−2 (1.73e−3)†	1.2190e−2 (1.82e−3)	DTLZ6	5.0035e−1 (5.09e−2)	4.9256e−1 (4.74e−2)
WFG3	1.2398e−2 (4.02e−4)†	1.2146e−2 (3.77e−4)	DTLZ7	2.2135e−1 (2.15e−1)†	6.2268e−2 (1.16e−3)
WFG4	1.2892e−2 (3.68e−4)‡	1.2945e−2 (2.50e−4)	UF8	1.4014e−1 (4.17e−2)	1.3654e−1 (4.47e−2)
WFG5	6.6388e−2 (2.12e−3)‡	6.6740e−2 (2.14e−4)	UF9	1.8481e−1 (4.28e−2)†	1.6542e−1 (5.21e−2)
WFG6	3.0332e−2 (2.42e−2)	2.6514e−2 (1.44e−2)	UF10	3.2988e−1 (5.51e−2)	3.1911e−1 (6.83e−2)

†The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is better than its competitor.
‡The p value of 98 DOF is significant at a .05 level of significance by two-tailed t-test. ETEA is worse than its competitor.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

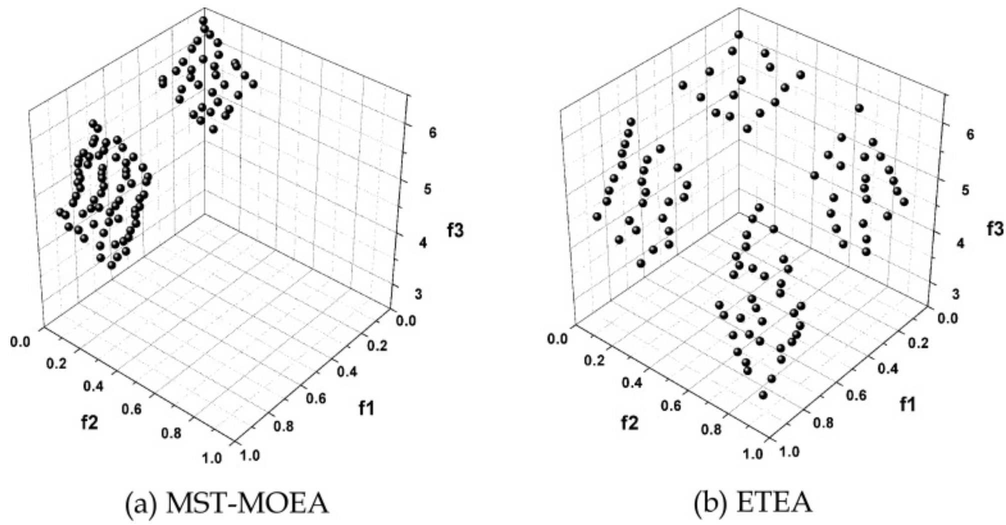

Figure 11: The final solutions obtained by MST-MOEA and ETEA on DTLZ7.

problems, and with statistical significance on 20 problems. Concerning IGD, it is able to achieve better results in all the problems except WFG4, WFG5, and WFG9, and with statistical significance on 21 problems. Additionally, for some problems, such as ZDT4 and DTLZ7, the MST-MOEA algorithm cannot always find the whole Pareto front. The frequency of success is 18/50 and 31/50 on ZDT4 and DTLZ7, respectively. The typical failure for DTLZ7 is shown in Figure 11, which may be attributed to the sensitivity of fitness to the density estimator in MST-MOEA: the fitness of some dominated individuals will be very poor if the estimation value of their corresponding non-dominated individuals is significantly low. However, these dominated individuals may be located in the outer part of the population and close to the boundary solutions, and thus can largely extend the search toward undeveloped areas.

5.4 Study of Different Components of ETEA

In the previous sections, we studied ETEA as a whole by comparing it with the other five state-of-the-art EMO algorithms and its predecessor. However, an investigation of the different components of the proposed algorithm is also very useful to allow us to understand whether these newly introduced components are indeed able to improve the algorithm as well as how these different components contribute to the performance of the algorithm. The main components of ETEA are the density estimator, fitness assignment, fitness adjustment, and archive truncation. Here, we investigate their contribution by simply exchanging them with the standard algorithm or by directly removing them.

NSGA-II is selected as the standard algorithm due to its reputation in the EMO community. ETEA separately exchanges the density estimator, fitness assignment, and archive truncation mechanisms with NSGA-II, called $\text{ETEA}_{(\text{EDE})}$, $\text{ETEA}_{(\text{EFA})}$, and $\text{ETEA}_{(\text{EAT})}$, respectively; the fitness adjustment mechanism in ETEA is removed since it is nonexistent in NSGA-II and the new resulting algorithm is called $\text{ETEA}_{(\text{RFA})}$. Table 11 gives the HV values of the four algorithms on all the test problems, and the results of the original ETEA are also repeated in the table. Additionally, for a clearer comparison, Table 11 shows the rank of the five algorithms on each problem.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 11: Comparison of the HV results on all the 32 problems. ETEA$_{(EDE)}$, ETEA$_{(EFA)}$, and ETEA$_{(EAT)}$ separately correspond to the new ETEA algorithms whose density estimator, fitness assignment, and archive truncation mechanisms come from NSGA-II. ETEA$_{(RFA)}$ stands for the new ETEA algorithm where the fitness adjustment mechanism is removed. The number in the upper right quadrant associated with an algorithm indicates its rank among the five algorithms for a test problem. The value in the last row means the sum rank for all the problems.

Problem	ETEA^{rank}	ETEA$_{(EDE)}{}^{rank}$	ETEA$_{(EFA)}{}^{rank}$	ETEA$_{(EAT)}{}^{rank}$	ETEA$_{(RFA)}{}^{rank}$
SCH1	2.2275e+1 $_{(6.47e-4)}{}^{1}$	2.2274e+1 $_{(6.82e-4)}{}^{4}$	2.2275e+1 $_{(4.62e-4)}{}^{1}$	2.2270e+1 $_{(1.54e-3)}{}^{5}$	2.2275e+1 $_{(4.18e-4)}{}^{1}$
SCH2	3.8259e+1 $_{(2.12e-3)}{}^{1}$	3.8258e+1 $_{(1.61e-3)}{}^{2}$	3.8258e+1 $_{(2.87e-3)}{}^{2}$	3.8247e+1 $_{(4.14e-3)}{}^{5}$	3.8258e+1 $_{(3.95e-3)}{}^{2}$
FON	3.0621e+0 $_{(1.80e-4)}{}^{2}$	3.0621e+0 $_{(1.39e-4)}{}^{2}$	3.0622e+0 $_{(9.32e-5)}{}^{1}$	3.0617e+0 $_{(2.13e-4)}{}^{5}$	3.0621e+0 $_{(1.44e-4)}{}^{2}$
KUR	3.7072e+1 $_{(1.05e-2)}{}^{2}$	3.7072e+1 $_{(1.02e-2)}{}^{2}$	3.7068e+1 $_{(1.05e-2)}{}^{4}$	3.7008e+1 $_{(1.53e-2)}{}^{5}$	3.7079e+1 $_{(5.36e-3)}{}^{1}$
POL	7.5316e+1 $_{(4.33e-2)}{}^{1}$	7.5015e+1 $_{(9.02e-2)}{}^{5}$	7.5250e+1 $_{(6.88e-2)}{}^{4}$	7.5268e+1 $_{(4.72e-2)}{}^{3}$	7.5306e+1 $_{(5.69e-2)}{}^{2}$
ZDT1	3.6601e+0 $_{(3.92e-4)}{}^{2}$	3.6604e+0 $_{(2.02e-4)}{}^{1}$	3.6597e+0 $_{(4.62e-4)}{}^{4}$	3.6586e+0 $_{(5.44e-4)}{}^{5}$	3.6600e+0 $_{(4.46e-4)}{}^{3}$
ZDT2	3.3260e+0 $_{(6.68e-4)}{}^{1}$	3.3259e+0 $_{(4.41e-4)}{}^{2}$	3.3257e+0 $_{(7.64e-4)}{}^{3}$	3.3242e+0 $_{(8.61e-4)}{}^{5}$	3.3248e+0 $_{(1.10e-3)}{}^{4}$
ZDT3	4.8131e+0 $_{(4.47e-4)}{}^{2}$	4.8130e+0 $_{(5.66e-4)}{}^{3}$	4.8124e+0 $_{(4.20e-4)}{}^{5}$	4.8125e+0 $_{(5.23e-4)}{}^{4}$	4.8132e+0 $_{(2.05e-4)}{}^{1}$
ZDT4	3.6514e+0 $_{(7.73e-3)}{}^{1}$	3.6512e+0 $_{(1.05e-2)}{}^{2}$	3.5987e+0 $_{(7.44e-2)}{}^{4}$	3.6507e+0 $_{(6.54e-3)}{}^{3}$	3.5865e+0 $_{(6.36e-2)}{}^{5}$
ZDT6	3.0242e+0 $_{(2.58e-3)}{}^{2}$	3.0247e+0 $_{(2.51e-3)}{}^{1}$	3.0196e+0 $_{(2.82e-3)}{}^{4}$	3.0226e+0 $_{(2.41e-3)}{}^{3}$	3.0172e+0 $_{(4.15e-3)}{}^{5}$
WFG1	7.2435e+0 $_{(1.26e+0)}{}^{2}$	7.7706e+0 $_{(9.66e-1)}{}^{1}$	6.7730e+0 $_{(7.00e-1)}{}^{5}$	7.2177e+0 $_{(9.06e-1)}{}^{3}$	6.9252e+0 $_{(9.12e-1)}{}^{4}$
WFG2	1.1151e+1 $_{(4.17e-1)}{}^{1}$	1.0870e+1 $_{(3.84e-1)}{}^{3}$	1.0868e+1 $_{(3.81e-1)}{}^{4}$	1.0702e+1 $_{(2.49e-1)}{}^{5}$	1.1035e+1 $_{(4.16e-1)}{}^{2}$
WFG3	1.0944e+1 $_{(5.11e-3)}{}^{1}$	1.0940e+1 $_{(1.12e-2)}{}^{3}$	1.0941e+1 $_{(4.78e-3)}{}^{2}$	1.0935e+1 $_{(8.28e-3)}{}^{5}$	1.0937e+1 $_{(6.17e-3)}{}^{4}$
WFG4	8.6679e+0 $_{(7.34e-3)}{}^{1}$	8.6672e+0 $_{(4.52e-3)}{}^{4}$	8.6671e+0 $_{(4.54e-3)}{}^{3}$	8.6564e+0 $_{(8.03e-3)}{}^{5}$	8.6678e+0 $_{(4.59e-3)}{}^{2}$
WFG5	8.1575e+0 $_{(3.00e-2)}{}^{3}$	8.1629e+0 $_{(3.16e-2)}{}^{2}$	8.1511e+0 $_{(2.33e-2)}{}^{5}$	8.1751e+0 $_{(3.29e-2)}{}^{1}$	8.1521e+0 $_{(2.98e-2)}{}^{4}$
WFG6	8.5708e+0 $_{(1.07e-1)}{}^{3}$	8.5490e+0 $_{(1.64e-1)}{}^{5}$	8.5815e+0 $_{(3.11e-2)}{}^{2}$	8.5701e+0 $_{(8.59e-2)}{}^{4}$	8.5891e+0 $_{(5.88e-2)}{}^{1}$
WFG7	8.6703e+0 $_{(6.64e-3)}{}^{2}$	8.6747e+0 $_{(2.31e-3)}{}^{1}$	8.6687e+0 $_{(4.39e-3)}{}^{3}$	8.6603e+0 $_{(4.90e-3)}{}^{5}$	8.6684e+0 $_{(5.33e-3)}{}^{4}$

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Table 11: Continued.

Problem	$\mathrm{ETEA}^{\mathrm{rank}}$	$\mathrm{ETEA}_{(\mathrm{EDE})}{}^{\mathrm{rank}}$	$\mathrm{ETEA}_{(\mathrm{EFA})}{}^{\mathrm{rank}}$	$\mathrm{ETEA}_{(\mathrm{EAT})}{}^{\mathrm{rank}}$	$\mathrm{ETEA}_{(\mathrm{RFA})}{}^{\mathrm{rank}}$
WFG8	7.0008e+0 $_{(3.61e-1)}{}^{2}$	7.1086e+0 $_{(4.13e-1)}{}^{1}$	6.9487e+0 $_{(3.63e-1)}{}^{5}$	7.0004e+0 $_{(4.30e-1)}{}^{3}$	6.9849e+0 $_{(4.59e-1)}{}^{4}$
WFG9	8.4377e+0 $_{(1.57e-2)}{}^{1}$	8.4339e+0 $_{(2.00e-2)}{}^{4}$	8.4367e+0 $_{(1.77e-2)}{}^{2}$	8.4334e+0 $_{(1.64e-2)}{}^{5}$	8.4345e+0 $_{(1.89e-2)}{}^{3}$
VNT1	6.1582e+1 $_{(4.15e-2)}{}^{2}$	6.1577e+1 $_{(2.59e-2)}{}^{3}$	6.1565e+1 $_{(3.62e-2)}{}^{4}$	6.1125e+1 $_{(6.63e-2)}{}^{5}$	6.1593e+1 $_{(3.32e-2)}{}^{1}$
VNT2	1.9146e+0 $_{(3.90e-4)}{}^{1}$	1.9146e+0 $_{(4.74e-4)}{}^{1}$	1.9145e+0 $_{(4.40e-4)}{}^{3}$	1.9097e+0 $_{(1.79e-3)}{}^{5}$	1.9145e+0 $_{(5.53e-4)}{}^{3}$
VNT3	2.8303e+1 $_{(1.09e-2)}{}^{2}$	2.8298e+1 $_{(9.26e-3)}{}^{4}$	2.8296e+1 $_{(1.01e-2)}{}^{5}$	2.8356e+1 $_{(3.73e-3)}{}^{1}$	2.8301e+1 $_{(7.94e-3)}{}^{3}$
DTLZ1	9.7286e−1 $_{(3.30e-4)}{}^{1}$	9.7237e−1 $_{(6.38e-4)}{}^{3}$	9.7246e−1 $_{(7.69e-4)}{}^{2}$	9.6844e−1 $_{(1.35e-3)}{}^{5}$	9.7233e−1 $_{(8.20e-4)}{}^{4}$
DTLZ2	7.3948e+0 $_{(6.57e-3)}{}^{2}$	7.3895e+0 $_{(6.26e-3)}{}^{4}$	7.3961e+0 $_{(6.61e-3)}{}^{1}$	7.3519e+0 $_{(1.83e-2)}{}^{5}$	7.3937e+0 $_{(6.44e-3)}{}^{3}$
DTLZ3	2.3436e+0 $_{(2.75e+0)}{}^{3}$	3.0471e+0 $_{(3.30e+0)}{}^{1}$	2.9068e+0 $_{(2.92e+0)}{}^{2}$	1.0205e+0 $_{(2.13e+0)}{}^{4}$	5.7624e−1 $_{(8.00e-1)}{}^{5}$
DTLZ4	7.2158e+0 $_{(3.19e-1)}{}^{1}$	7.1972e+0 $_{(5.62e-1)}{}^{4}$	7.1977e+0 $_{(4.79e-1)}{}^{3}$	7.1631e+0 $_{(2.01e-1)}{}^{5}$	7.2033e+0 $_{(4.04e-1)}{}^{2}$
DTLZ5	6.1015e+0 $_{(6.15e-4)}{}^{1}$	6.1015e+0 $_{(7.01e-4)}{}^{1}$	6.1015e+0 $_{(4.56e-4)}{}^{1}$	6.0994e+0 $_{(7.02e-4)}{}^{5}$	6.1013e+0 $_{(3.04e-4)}{}^{4}$
DTLZ6	4.6522e+0 $_{(1.66e-1)}{}^{1}$	4.1209e+0 $_{(2.86e-1)}{}^{5}$	4.3871e+0 $_{(1.88e-1)}{}^{4}$	4.5087e+0 $_{(1.77e-1)}{}^{3}$	4.6018e+0 $_{(1.71e-1)}{}^{2}$
DTLZ7	1.3400e+1 $_{(2.60e-2)}{}^{1}$	1.3366e+1 $_{(1.77e-2)}{}^{2}$	1.2778e+1 $_{(1.23e+0)}{}^{4}$	1.3082e+1 $_{(5.89e-2)}{}^{3}$	1.2473e+1 $_{(1.99e+0)}{}^{5}$
UF8	6.9641e+0 $_{(3.83e-1)}{}^{1}$	6.8878e+0 $_{(3.31e-1)}{}^{4}$	6.9029e+0 $_{(3.36e-1)}{}^{3}$	6.8279e+0 $_{(3.55e-1)}{}^{5}$	6.9633e+0 $_{(4.12e-1)}{}^{2}$
UF9	7.1990e+0 $_{(3.41e-1)}{}^{1}$	7.1382e+0 $_{(3.22e-1)}{}^{2}$	7.0438e+0 $_{(2.34e-1)}{}^{4}$	6.8085e+0 $_{(4.87e-1)}{}^{5}$	7.0967e+0 $_{(3.82e-1)}{}^{3}$
UF10	5.2946e+0 $_{(7.31e-1)}{}^{4}$	5.5982e+0 $_{(8.91e-1)}{}^{1}$	5.2723e+0 $_{(4.67e-1)}{}^{5}$	5.3518e+0 $_{(8.09e-1)}{}^{3}$	5.5171e+0 $_{(6.10e-1)}{}^{2}$
Sum Rank	52	83	104	132	93

Clearly, all the four components play an important role in ETEA. The algorithm with them exchanged with NSGA-II or removed performs worse for the majority of the test problems. As can be seen from the table, the sum rank of the original ETEA is significantly better than that of the other four algorithms (52 for ETEA against 83 for $\text{ETEA}_{(\text{EDE})}$, 104 for $\text{ETEA}_{(\text{EFA})}$, 132 for $\text{ETEA}_{(\text{EAT})}$, and 93 for $\text{ETEA}_{(\text{RFA})}$, respectively). On the other hand, considering the contribution of the different components to the algorithm, the archive truncation appears to be most influential to the performance of ETEA, followed by the fitness assignment, fitness adjustment, and density estimation (the sum rank of $\text{ETEA}_{(\text{EDE})}$, $\text{ETEA}_{(\text{EFA})}$, $\text{ETEA}_{(\text{EAT})}$, and $\text{ETEA}_{(\text{RFA})}$ is 132, 104, 93, and 83, respectively). This means that a good archive truncation strategy which can effectively maintain both uniformity and spread of a population is very important in an EMO algorithm.

6 Conclusions and Future Work

This paper proposes a Euclidean minimum spanning tree-based evolutionary algorithm, denoted ETEA, to solve multi-objective optimization problems. ETEA explores the characteristics of EMSTs and the distance relation among individuals to guide the search during the evolutionary process. On the one hand, ETEA defines a density estimator ETCD which not only reflects the crowding degree of individuals but partly implies their relative orientation and position in the population, and combines it with several properties of EMSTs to maintain diversity in fitness assignment and archive truncation. On the other hand, ETEA considers the neighborhood of individuals formed by their distance to the obtained trade-off surface, and adjusts their fitness according to the number and position of individuals in the neighborhood.

Systematic experiments have been performed by making an extensive comparison of ETEA with five state-of-the-art EMO algorithms (NSGA-II, SPEA2, IBEA, ϵ-MOEA, and TDEA), and its predecessor (MST-MOEA). Thirty-two test problems and two popular quality metrics are chosen to assess performance of the selected algorithms. The results reveal that ETEA can provide a good balance in finding a well-approximated non-dominated set, keeping the uniformity of solutions, and extending the distribution range along the optimal front. ETEA performs significantly better than the other algorithms in terms of uniformity, and is competitive in convergence and spread. The results of ETEA with respect to the comprehensive performance also achieve the best among the tested algorithms for the majority of the test problems.

Furthermore, the contribution of different components of the proposed algorithm has been experimentally investigated. The results show that the newly introduced archive truncation is most influential to the performance of ETEA, followed by the fitness assignment, fitness adjustment, and density estimation. In addition, the study on runtime indicates that although ETEA generally requires more computational time than the other algorithms, it can be improved by several efficient methods.

One area for subsequent work is to obtain a deeper understanding of the algorithm behavior. In this context, the effects of the population size and the number of variables will be investigated. In addition, applying ETEA to some constrained MOPs and real-world scenarios is also an important aspect of our further research.

Acknowledgments

The authors would like to thank the anonymous Associate Editor and reviewers for their thoughtful suggestions and constructive comments. The authors would also like to acknowledge the help of Dr. Aimin Zhou and Dr. Yong Wang on this work. The work was supported by the

Engineering and Physical Sciences Research Council (EPSRC) of the United Kingdom under Grant EP/K001310/1, the National Natural Science Foundation of China under Grant 61070088, a Ph.D. Studentship from the School of Information Systems, Computing, and Mathematics, Brunel University, and the Education Department of Jiangsu, China.

References

Abraham, A., Jain, L., and Goldberg, R. (Eds.). (2005). *Evolutionary multiobjective optimization: Theoretical advances and applications*. Berlin: Springer.

Agarwal, P. K., Edelsbrunner, H., Schwarzkopf, O., and Welzl, E. (1991). Euclidean minimum spanning trees and bichromatic closest pairs. *Discrete and Computational Geometry*, 6(1):407–422.

Auger, A., Bader, J., Brockhoff, D., and Zitzler, E. (2009). Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point. In *Proceedings of the 10th ACM SIGEVO Workshop on Foundations of Genetic Algorithms, FOGA '09*, pp. 87–102.

Bader, J., and Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary Computation*, 19(1):45–76.

Bansal, T., and Ghanshani, P. (2006). An EMST based look-up protocol for peer to peer networks. *Journal of Networks*, 1(3):9–17.

Beume, N., Naujoks, B., and Emmerich, M. (2007). SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669.

Bosman, P. A. N., and Thierens, D. (2003). The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):174–188.

Branke, J., Deb, K., Miettinen, K., and Slowinski, R. (Eds.). (2008). *Multiobjective optimization: Interactive and evolutionary approaches*. Berlin: Springer-Verlag.

Brockhoff, D., Friedrich, T., Hebbinghaus, N., Klein, C., Neumann, F., and Zitzler, E. (2009). On the effects of adding objectives to plateau functions. *IEEE Transactions on Evolutionary Computation*, 13(3):591–603.

Bui, L. T., and Alam, S. (Eds.). (2008). *Multi-objective optimization in computational intelligence: Theory and practice.* Hershey, PA: Information Science Reference.

Coello, C. A. C. (2011). Evolutionary multi-objective optimization. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(5):444–447.

Coello, C. A. C., and Lamont, G. B. (Eds.). (2004). *Applications of multi-objective evolutionary algorithms*. Singapore: World Scientific.

Coello, C. A. C., Veldhuizen, D. A. V., and Lamont, G. B. (2007). *Evolutionary algorithms for solving multi-objective problems*, 2nd ed. Berlin: Springer.

Corne, D. W., Jerram, N. R., Knowles, J. D., and Oates, M. J. (2001). PESA-II: Region-based selection in evolutionary multiobjective optimization. In *Proceedings of the 2001 Genetic and Evolutionary Computation Conference, GECCO 2001*, pp. 283–290.

Czumaj, A., Ergün, F., Fortnow, L., Magen, A., Newman, I., Rubinfeld, R., and Sohler, C. (2003). Sublinear-time approximation of Euclidean minimum spanning tree. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 813–822.

Czumaj, A., Ergün, F., Fortnow, L., Magen, A., Newman, I., Rubinfeld, R., and Sohler, C. (2005). Approximating the weight of the Euclidean minimum spanning tree in sublinear time. *SIAM Journal on Computing*, 35(1):91–109.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. New York: John Wiley.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Deb, K., Mohan, M., and Mishra, S. (2005). Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions. *Evolutionary Computation*, 13(4):501–525.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197.

Deb, K., Sinha, A., and Kukkonen, S. (2006). Multi-objective test problems, linkages, and evolutionary methodologies. In *Proceedings of the 2006 Genetic and Evolutionary Computation Conference, GECCO 2006*, pp. 1141–1148.

Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In A. Abraham, L. Jain, and R. Goldberg (Eds.), *Evolutionary multiobjective optimization: Theoretical advances and applications* (pp. 105–145). Berlin: Springer.

Durillo, J. J., Nebro, A. J., Luna, F., and Alba, E. (2009). On the effect of the steady-state selection scheme in multi-objective genetic algorithms. In *Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009). Lecture notes in computer science*, Vol. 5467 (pp. 183–197). Berlin: Springer-Verlag.

Elhossini, A., Areibi, S., and Dony, R. (2010). Strength Pareto particle swarm optimization and hybrid EA-PSO for multi-objective optimization. *Evolutionary Computation*, 18(1):127–156.

Eppstein, D. (1995). Dynamic Euclidean minimum spanning trees and extrema of binary functions. *Discrete and Computational Geometry*, 13(1):111–122.

Farhang-Mehr, A., and Azarm, S. (2002). Diversity assessment of Pareto-optimal solution sets: An entropy approach. In *Proceedings of the 2002 IEEE World Congress on Computational Intelligence*, pp. 723–728.

Farina, M., and Amato, P. (2003). Fuzzy optimality and evolutionary multiobjective optimization. In *Proceedings of the 2nd International Conference on Evolutionary Multi-Criterion Optimization, EMO 2003. Lecture notes in computer science*, Vol. 2632 (pp. 58–72). Berlin: Springer-Verlag.

Fonseca, C. M., and Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3(1):1–16.

Friedrich, T., He, J., Hebbinghaus, N., Neumann, F., and Witt, C. (2010). Approximating covering problems by randomized search heuristics using multi-objective models. *Evolutionary Computation*, 18(4):617–633.

Gong, M., Jiao, L., Du, H., and Bo, L. (2008). Multiobjective immune algorithm with nondominated neighbor-based selection. *Evolutionary Computation*, 16(2):225–255.

Hanne, T. (2001). Global multiobjective optimization with evolutionary algorithms: Selection mechanisms and mutation control. In *Proceedings of the 1st International Conference on Evolutionary Multi-Criterion Optimization, EMO 2001. Lecture notes in computer science*, Vol. 1993 (pp. 197–212). Berlin: Springer-Verlag.

Hernández-Díaz, A. G., Santana-Quintero, L. V., Coello Coello, C. A., and Molina, J. (2007). Pareto-adaptive ϵ-dominance. *Evolutionary Computation*, 15:493–517.

Horn, J., Nafpliotis, N., and Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, pp. 82–87.

Horoba, C., and Neumann, F. (2010). Approximating Pareto-optimal sets using diversity strategies in evolutionary multi-objective optimization. In C. C. A. Coello, C. Dhaenens, and L. Jourdan (Eds.), *Advances in multi-objective nature inspired computing studies in computational intelligence (SCI)*, Vol. 272 (pp. 23–44). Berlin: Springer-Verlag.

Huband, S., Hingston, P., Barone, L., and While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506.

Igel, C., Hansen, N., and Roth, S. (2007). Covariance matrix adaptation for multi-objective optimization. *Evolutionary Computation*, 15(1):1–28.

Igel, C., Suttorp, T., and Hansen, N. (2007). Steady-state selection and efficient covariance matrix update in the multi-objective CMA-ES. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007. Lecture notes in computer science*, Vol. 4403 (pp. 171–185). Berlin: Springer-Verlag.

Ikeda, K., Kita, H., and Kobayashi, S. (2001). Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal? In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 957–962.

Ishibuchi, H., Tsukamoto, N., and Nojima, Y. (2008). Evolutionary many-objective optimization: A short review. In *Proceedings of the 2008 IEEE Congress on Evolutionary Computation*, pp. 2419–2426.

Jin, Y. (Ed.). (2006). *Multi-objective machine learning*. Berlin: Springer.

Karahan, I., and Köksalan, M. (2010). A territory defining multiobjective evolutionary algorithm and preference incorporation. *IEEE Transactions on Evolutionary Computation*, 14(4):636–664.

Khor, E., Tan, K. C., Lee, T. H., and Goh, C. K. (2005). A study on distribution preservation mechanism in evolutionary multi-objective optimization. *Artificial Intelligence Review*, 23(1):31–56.

Knowles, J. D. (2006). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 10(1):50–66.

Knowles, J. D., and Corne, D. W. (2000). Approximating the nondominated front using the Pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172.

Knowles, J. D., and Corne, D. W. (2002). On metrics for comparing nondominated sets. In *Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, Vol. 1, pp. 711–716.

Kuhn, H. W., and Tucker, A. W. (1951). Nonlinear programming. In *Proceedings of the 2nd Berkeley Symposium on Mathematical Statistics and Probability*, pp. 481–492.

Kukkonen, S., and Deb, K. (2006). A fast and effective method for pruning of non-dominated solutions in many-objective problems. In *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX). Lecture notes in computer science*, Vol. 4193 (pp. 553–562). Berlin: Springer-Verlag.

Kumar, R., and Rockett, P. (2002). Improved sampling of the Pareto-front in multiobjective genetic optimizations by steady-state evolution: A Pareto converging genetic algorithm. *Evolutionary Computation*, 10(3):283–314.

Laumanns, M., Thiele, L., Deb, K., and Zitzler, E. (2002). Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*, 10(3):263–282.

Lee, I. K. (1999). Curve reconstruction from unorganized points. *Computer Aided Geometric Design*, 17(2):161–177.

Li, M., Yang, S., Liu, X., and Shen, R. (2013). A comparative study on evolutionary algorithms for many-objective optimization. In *Proceedings of the 7th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2013). Lecture notes in computer science*, Vol. 7811 (pp. 261–275). Berlin: Springer-Verlag.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Li, M., and Zheng, J. (2009). Spread assessment for evolutionary multi-objective optimization. In *Proceedings of the 5th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2009. Lecture notes in computer science*, Vol. 5476 (pp. 216–230). Berlin: Springer-Verlag.

Li, M., Zheng, J., Li, K., Wu, J., and Xiao, G. (2009). An spanning tree based method for pruning non-dominated solutions in multi-objective optimization problems. In *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, SMC 2009*, pp. 4882–4887.

Li, M., Zheng, J., Shen, R., Li, K., and Yuan, Q. (2010). A grid-based fitness strategy for evolutionary many-objective optimization. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation, GECCO 2010*, pp. 463–470.

Li, M., Zheng, J., and Xiao, G. (2008). An efficient multi-objective evolutionary algorithm based on minimum spanning tree. In *Proceedings of the 2008 IEEE Congress on Evolutionary Computation, CEC 2008*, pp. 617–624.

Li, X. (2003). A non-dominated sorting particle swarm optimizer for multiobjective optimization. In *Proceeding of the 2003 Genetic and Evolutionary Computation Conference, GECCO 2003*, pp. 37–48.

Nebro, A. J., Luna, F., Alba, E., Dorronsoro, B., Durillo, J. J., and Beham, A. (2008). AbYSS: Adapting scatter search to multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 12(4):439–457.

Okabe, T., Jin, Y., and Sendhoff, B. (2003). A critical survey of performance indices for multi-objective optimisation. In *Proceedings of the 2003 IEEE Congress Evolutionary Computation, CEC 2003*, Vol. 2, pp. 878–885.

Purshouse, R. C., and Fleming, P. J. (2007). On the evolutionary optimization of many conflicting objectives. *IEEE Transactions on Evolutionary Computation*, 11(6):770–784.

Rajasekaran, S. (2005). On the Euclidean minimum spanning tree problem. *Computing Letters*, 1(1):11–14.

Schott, J. R. (1995). Fault tolerant design using single and multicriteria genetic algorithm optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA.

Schütze, O., Lara, A., and Coello, C. A. C. (2011). On the influence of the number of objectives on the hardness of a multiobjective optimization problem. *IEEE Transactions on Evolutionary Computation*, 15(4):444–455.

Sen, P., and Yang, J. (1998). *Multiple criteria decision support in engineering design*. Berlin: Springer.

Shir, O. M., Emmerich, M., and Bäck, T. (2010). Adaptive niche radii and niche shapes approaches for niching with the CMA-ES. *Evolutionary Computation*, 18(1):97–126.

Tan, K. C., Khor, E. F., and Lee, T. H. (2005). Multiobjective evolutionary algorithms and applications. Berlin: Springer-Verlag.

Tan, K. C., Lee, T. H., and Khor, E. F. (2001). Evolutionary algorithms for multi-objective optimization: Performance assessments and comparisons. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, Vol. 2, pp. 979–986.

Teo, J., and Abbass, H. A. (2004). Automatic generation of controllers for embodied legged organisms: A Pareto evolutionary multi-objective approach. *Evolutionary Computation*, 12(12):355–394.

Van Veldhuizen, D. A. (1999). Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations. Ph.D. thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Kettering, Ohio.

Evolutionary Computation corrected proof
doi:10.1162/EVCO_a_00106

Van Veldhuizen, D. A., and Lamont, G. B. (1998). Evolutionary computation and convergence to a Pareto front. In *Late Breaking Papers at the Genetic Programming Conference*, pp. 221–228.

Šeda, M. (2008). Geometric data structures and their selected applications,. *International Journal of Computer Systems Science and Engineering*, 3(4):168–173.

Wagner, T., Beume, N., and Naujoks, B. (2007). Pareto-, aggregation-, and indicator-based methods in many-objective optimization. In *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization, EMO 2007. Lecture notes in computer science*, Vol. 4403 (pp. 742–756). Berlin: Springer-Verlag.

Wang, Z., Tang, K., and Yao, X. (2010). Multi-objective approaches to optimal testing resource allocation in modular software systems. *IEEE Transactions on Reliability*, 59(3):563–575.

Wei, J., and Zhang, M. (2011). A memetic particle swarm optimization for constrained multi-objective optimization problems. In *Proceedings of the 2011 IEEE Congress on Evolutionary Computation, CEC 2011*, pp. 1636–1643.

Wieland, S. C., Brownstein, J. S., Berger, B., and Mandl, K. D. (2007). Density-equalizing Euclidean minimum spanning trees for the detection of all disease cluster shapes. In *Proceedings of the National Academy of Sciences of the United States of America*, 104:9404–9409.

Yang, S., Li, M., Liu, X., and Zheng, J. (2013). A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, Doi:10:1109/TEVC.2012.2227145

Yen, G. G., and Lu, H. (2003). Dynamic multiobjective evolutionary algorithm: Adaptive cell-based rank and density estimation. *IEEE Transactions on Evolutionary Computation*, 7(3):253–274.

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

Zhang, Q., Zhou, A., and Jin, Y. (2008). RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 12(1):41–63.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., and Tiwari, S. (2009). Multiobjective optimization test instances for the CEC 2009 special session and competition. School of CS & EE, University of Essex. Working Report CES-487.

Zitzler, E., Deb, K., and Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195.

Zitzler, E., and Künzli, S. (2004). Indicator-based selection in multiobjective search. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII). Lecture notes in computer science*, Vol. 3242 (pp. 832–842). Berlin: Springer-Verlag.

Zitzler, E., Laumanns, M., and Bleuler, S. (2004). A tutorial on evolutionary multiobjective optimization. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt (Eds.), *Metaheuristics for multiobjective optimisation* (pp. 3–37). Berlin: Springer.

Zitzler, E., Laumanns, M., and Thiele, L. (2002). SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Proceedings of the 4th International Conference on Evolutionary Methods for Design, Optimisation and Control with Applications to Industrial Problems, EUROGEN2001*, pp. 95–100.

Zitzler, E., and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271.

Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., and da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132.