Information Sciences 262 (2014) 62–77

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Runtime analysis of a multi-objective evolutionary algorithm for obtaining finite approximations of Pareto fronts

Yu Chen [a], Xiufen Zou [b,*]

[a] School of Science, Wuhan University of Technology, Wuhan 430070, China
[b] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China

ARTICLE INFO

Article history:
Received 5 June 2012
Received in revised form 12 June 2013
Accepted 19 November 2013
Available online 27 November 2013

Keywords:
Runtime analysis
Multi-objective evolutionary algorithm
Finite approximations of Pareto fronts

ABSTRACT

Previous theoretical analyses of evolutionary multi-objective optimization (EMO) mostly focus on obtaining ϵ-approximations of Pareto fronts. However, in practical applications, an appropriate value of ϵ is critical but sometimes, for a multi-objective optimization problem (MOP) with unknown attributes, difficult to determine. In this paper, we propose a new definition for the finite representation of the Pareto front—the adaptive Pareto front, which can automatically accommodate the Pareto front. Accordingly, it is more practical to take the adaptive Pareto front, or its ϵ-approximation (termed the ϵ-adaptive Pareto front) as the goal of an EMO algorithm. We then perform a runtime analysis of a $(\mu+1)$ multi-objective evolutionary algorithm ($(\mu+1)$ MOEA) for three MOPs, including a discrete MOP with a polynomial Pareto front (denoted as a polynomial DMOP), a discrete MOP with an exponential Pareto front (denoted as an exponential DMOP) and a simple continuous two-objective optimization problem (SCTOP). By employing an estimator-based update strategy in the $(\mu+1)$ MOEA, we show that (1) for the polynomial DMOP, the whole Pareto front can be obtained in the expected polynomial runtime by setting the population size μ equal to the number of Pareto vectors; (2) for the exponential DMOP, the expected polynomial runtime can be obtained by keeping μ increasing in the same order as that of the problem size n; and (3) the diversity mechanism guarantees that in the expected polynomial runtime the MOEA can obtain an ϵ-adaptive Pareto front of SCTOP for any given precision ϵ. Theoretical studies and numerical comparisons with NSGA-II demonstrate the efficiency of the proposed MOEA and should be viewed as an important step toward understanding the mechanisms of MOEAs.

1. Introduction

Recently, various soft computing techniques have been widely utilized in the fields of science and engineering [30,37,9,22,36,39]. One set of powerful soft computing method is multi-objective evolutionary algorithms (MOEAs). These algorithms can explore the feasible spaces of multi-objective optimization problems (MOPs) to obtain uniformly distributed Pareto vectors, which has been shown by abundant numerical results [41,24,42,10,43,21,11,2,38,44,7,13,23,29,34,35]. Meanwhile, theoretical studies of convergence [26,25,16,40,32,8,1] and runtime analyses [14,26,28,31,5,6,18,19,3,15,20,4,12,33] have also been performed to explain how MOEAs function on different MOPs.

Laumanns et al. [27,28] investigated the "leading ones, trailing zeros" (LOTZ) problem and demonstrated that the expected runtime of the simple evolutionary multi-objective optimizer (SEMO) for LOTZ is $\Theta(n^3)$. Giel [14] extended the runtime analysis to the Global SEMO (GSEMO) by investigating the LOTZ problem and another simple test problem, and

* Corresponding author.
E-mail addresses: xfzou@whu.edu.cn, zouxiufen@yahoo.com (X. Zou).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.11.023

Neumann [31] found that the GSEMO can accommodate the Pareto front of a multi-objective minimum-spanning tree problem in the expected pseudo-polynomial runtime if the Pareto front is strongly convex. Moreover, Horoba [20] showed that the diversity-maintaining evolutionary multi-objective optimizer (DEMO) is a fully polynomial-time randomized approximation scheme for multi-objective shortest path problems. To theoretically confirm the efficiencies of hypervolume-based MOEAs, Beume et al. [3] compared the individual-based $\mathcal{S}$ metric selection evolutionary multi-objective optimization algorithm (SMS-EMOA) with the single-individual models of the nondominated sorting genetic algorithm II (NSGA-II) and the improved strength Pareto evolutionary algorithm (SPEA2), and then investigated the convergence rates of several population-based variants of SMS-EMOA [4].

By adding objectives to a well-known plateau function, Brockhoff et al. [6] found that changes in running time are caused by changes in the dominance structure. Subsequently, Schütze et al. [33] demonstrated that even if an increase in the number of objectives makes the problem more difficult, this increase in difficulty is sometimes not significant. Moreover, Laumanns *et al.* [28] verified the population's beneficial function through rigorous runtime analyses, while Giel and Lehre [15] further declared that there could be an exponential runtime gap between the population-based algorithms and single individual-based algorithms.

To understand the convergence properties of population-based MOEAs more concretely, Brockhoff et al. [5] analyzed the hypervolume-based MOEAs and obtained a polynomial upper bound on the expected runtime—to obtain an ϵ-approximation of an exponentially large Pareto front. By analyzing the runtime behaviors of MOEAs employing different diversity-preserving mechanisms, Friedrich et al. [12] demonstrated that certain mechanisms can improve the efficiencies of MOEAs on certain MOPs. Meanwhile, Horoba and Neumann [18,19] proposed several sufficient conditions for obtaining ϵ-Pareto sets of some MOPs they investigated. The theoretical results showed that although an ϵ-dominance approach can help achieve a good approximation for a Pareto set for some MOPs, this approach sometimes prevents the population from distributing uniformly along a small Pareto front. However, an MOEA based on a density estimator performs well in this case.

Existing theoretical results on runtime analysis have generally focused on dominance- or indicator-based MOEAs that were employed to obtain an ϵ-Pareto front of an MOP. To obtain an ϵ-Pareto front, the population size μ must be greater than or equal to a given threshold M, and the case where $\mu < M$ has not yet been considered. For a given precision ϵ, it is hardly feasible to choose a proper population size μ when an MOP with unknown attributes is encountered, whereas a large population will lead to high computation complexity and a small approximate Pareto front cannot represent the whole Pareto front precisely. By incorporating a fitness function compatible with the dominance relation in a $(\mu + 1)$ MOEA, we take a so-called adaptive Pareto front [8] as the destination of population evolution, which can automatically accommodate the true Pareto front. Compared with NSGA-II and SPEA2, the $(\mu + 1)$ MOEA employs a strategy of population update based on a fitness function, by which the selection pressure can be greatly improved when applied to many-objective evolutionary problems. It can also eliminate the essential difficulty of the multi-objective evolutionary algorithm based on decomposition (MOEA/D), that is, the difficulty of generating a uniformly-distributed vector set guiding the evolution of the population. Then, we estimate the expected runtime of a $(\mu + 1)$ MOEA for obtaining adaptive Pareto fronts or ϵ-adaptive Pareto fronts of MOPs. The major contributions of this paper include:

- We take the adaptive Pareto front as the destination of population evolution, and in this way, eliminate the difficult task of selecting a rational population size for a given precision ϵ.
- We theoretically demonstrate that if the $(\mu + 1)$ MOEA is utilized to solve a discrete MOP with polynomial Pareto vectors (the LOTZ), it is more efficient to set the population size equal to the number of Pareto vectors rather than employ a small population to obtain a uniform representation of the Pareto front.
- For a discrete MOP, when the number of Pareto vectors is of exponential order (the LF'_δ), the universal upper bound of the expected runtime is also exponential. However, a polynomial increase in the expected runtime can also be obtained by setting $k - 1 < \frac{n}{\mu} \leqslant k$ for a given positive constant k, where n is the problem size and μ is the population size.
- We demonstrate that a $(\mu + 1)$ MOEA based on a density estimator is a good solver for an MOP with a Pareto front that is a continuous curve because, for any $\varepsilon > 0$, it can obtain an ε-approximation of the adaptive Pareto front in the expected polynomial runtime.
- By comparing a variant of the proposed $(\mu + 1)$ MOEA, termed the $(\mu + \mu)$ MOEA, with NSGA-II, we also show that the proposed method is competitive with some existing MOEAs.

The remainder of this paper is organized as follows. Section 2 introduces some preliminaries on MOPs and MOEAs, and in Section 3, we perform the runtime analysis of the proposed $(\mu + 1)$ MOEA for the three MOPs under investigation. To demonstrate the efficiency of the newly proposed MOEA, we compare numerical results with the NSGA-II in Section 4. Finally, Section 5 concludes the paper and presents future work to be carried out.

2. Preliminaries

2.1. Multi-objective optimization problems

In general, an MOP with m objectives is described as

$$\max F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})), \tag{1}$$

where $\mathbf{x} = (x_1, \cdots, x_n) \in S_x \subseteq \mathbb{R}^n$, $\mathbf{u} = (u_1, \cdots, u_m) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x})) \in S_y \subseteq \mathbb{R}^m$. S_x, the set of all feasible solutions, is called the *feasible region*, and $S_y = F(S_x)$ is called the *objective region*. When the design variables are real-valued, the problem is called a *continuous multi-objective optimization problem (CMOP)*; however, if the variables are restricted to discrete values, the MOP is called a *discrete multi-objective optimization problem (DMOP)*. The optimal solutions of an MOP are the so-called **Pareto solutions**.

Definition 1. Let $\mathbf{u} = (u_1, \ldots, u_m)$ and $\mathbf{v} = (v_1, \ldots, v_m)$ be two vectors in the objective region S_y of MOP (1).

1. **(Pareto Dominance)** $\mathbf{u}$ is said to **Pareto dominate** $\mathbf{v}$, denoted as $\mathbf{u} \succ \mathbf{v}$, if and only if
   (a) **u weakly dominates v** (denoted as $\mathbf{u} \succeq \mathbf{v}$), i.e., $\forall\, i \in \{1, \ldots, m\} : u_i \geqslant v_i$;
   (b) $\exists\, j \in \{1, \ldots, m\} : u_j > v_j$.
2. **(Pareto Front & Pareto Set)**
   (a) A vector $\mathbf{u} \in S_y$ is called a **Pareto vector** of MOP (1) if there exists no $\mathbf{v} \in S_y$ satisfying $\mathbf{v} \succ \mathbf{u}$. The set of all Pareto vectors of MOP (1) is called the **Pareto front** of MOP (1), denoted as *PF*;
   (b) A feasible solution $\mathbf{x} \in S_x$ is called a **Pareto solution** of MOP (1) if $F(\mathbf{x})$ is a Pareto vector of MOP (1). All Pareto solutions of MOP (1) constitute the **Pareto set** of MOP (1), denoted as *PS*.

Sometimes, there are several feasible solutions corresponding to a common objective vector, in which case they are called **indifferent**. Indifferent solutions can be represented by their common objective vector. Therefore, our goals are to achieve reasonable approximations of the Pareto fronts and to estimate the expected runtime required for MOEAs to obtain such approximations. For a CMOP, the total number of Pareto vectors is usually uncountable, while the number of Pareto vectors for DMOPs is often finite.

Definition 2. Denote n to be the number of decision variables of an MOP. According to the number of Pareto vectors, DMOPs can be divided into two different categories:

1. **polynomial DMOPs,** where the number of Pareto vectors is $\mathcal{O}(n^k), k \in \mathbb{Z}^+$;
2. **exponential DMOPs,** where the number of Pareto vectors is $\Omega(k^n), k > 1$.

In the following, three MOPs are investigated to demonstrate the efficiency of the proposed MOEA.

2.1.1. The LOTZ problem

The LOTZ problem is a polynomial DMOP defined as

$$\max\ LOTZ(\mathbf{x}) = (LOTZ_1(\mathbf{x}), LOTZ_2(\mathbf{x})) = \left(\sum_{i=1}^{n}\prod_{j=1}^{i} x_j, \sum_{i=1}^{n}\prod_{j=i}^{n}(1 - x_j)\right),$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \{0, 1\}^n$. According to the sum of both objective values, the objective region of LOTZ can be partitioned into $n + 1$ sets $F_i, i = 0, 1, \ldots, n$, where the index i corresponds to the sum of both objectives (see Fig. 1). Obviously, $F_n = \emptyset$, and $F_n + 1$ is the Pareto front [28].

2.1.2. The LF'_δ problem

Let $\mathbf{x} = (x_1, \ldots, x_n)$ be a binary string, and assume that n is even. We denote the first half of $\mathbf{x}$ by $\ell(\mathbf{x}) = (x_1, \ldots, x_{n/2})$, and we denote its second half by $\hbar(\mathbf{x}) = (x_{n/2+1}, \ldots, x_n)$. For a bit string $\mathbf{b}$, we denote its length by $|\mathbf{b}|$, the number of 1-bits by $|\mathbf{b}|_1$, and its complement by $\bar{\mathbf{b}}$. Then, the real value of a bit-string $\mathbf{x}$ is

$$BV(\mathbf{x}) = \sum_{i=1}^{|\mathbf{x}|} 2^{|\mathbf{x}|-i} \cdot x_i,$$

and the LF'_δ problem [19] is described as

$$\max\ LF'_\delta(\mathbf{x}) = \left(LF'_{\delta,1}(\mathbf{x}), LF'_{\delta,2}(\mathbf{x})\right),$$

where

$$LF'_{\delta,1}(\mathbf{x}) := \begin{cases} (2 \cdot |\ell(\mathbf{x})|_1 + 2^{-n/2} \cdot BV(\hbar(\mathbf{x}))) \cdot \delta & \min\{|\ell(\mathbf{x})|_1, |\overline{\ell(\mathbf{x})}|_1\} \geqslant \sqrt{n}, \\ 2 \cdot |\ell(\mathbf{x})|_1 \cdot \delta & \text{otherwise}; \end{cases}$$

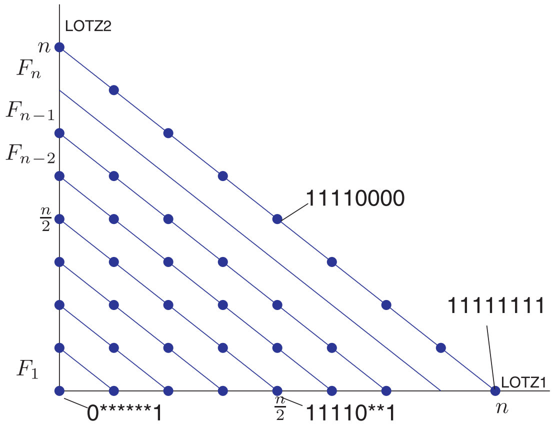

Fig. 1. Objective space of the LOTZ problem for $n = 8$.

$$LF'_{\delta,2}(\mathbf{x}) := \begin{cases} (2 \cdot |\overline{\ell(\mathbf{x})}|_1 + 2^{-n/2} \cdot BV(\overline{\hbar(\mathbf{x})})) \cdot \delta & \min\{|\ell(\mathbf{x})|_1, |\overline{\ell(\mathbf{x})}|_1\} \geqslant \sqrt{n}, \\ 2 \cdot |\overline{\ell(\mathbf{x})}|_1 \cdot \delta & \text{otherwise}. \end{cases}$$

All feasible solutions of the LF'_δ problem are Pareto solutions. When $\min\{|\ell(\mathbf{x})|_1, |\overline{\ell(\mathbf{x})}|_1\} < \sqrt{n}$, then $\binom{n/2}{|\ell(\mathbf{x})|_1} \cdot 2^{n/2}$ feasible solutions with the same value of $|\ell(\mathbf{x})|_1$ are mapped to the common Pareto vector $(2 \cdot |\ell(\mathbf{x})|_1 \cdot \delta, 2 \cdot |\overline{\ell(\mathbf{x})}|_1 \cdot \delta)$; otherwise, a Pareto vector $((2 \cdot |\ell(\mathbf{x})|_1 + 2^{-n/2} \cdot BV(\hbar(\mathbf{x}))) \cdot \delta, (2 \cdot |\overline{\ell(\mathbf{x})}|_1 + 2^{-n/2} \cdot BV(\overline{\hbar(\mathbf{x})})) \cdot \delta)$ is the image of $\binom{n/2}{|\ell(\mathbf{x})|_1}$ decision vectors that have the same values of $|\ell(\mathbf{x})|_1$ and $BV(\hbar(\mathbf{x}))$. Thus, LF'_δ is an exponential DMOP with a Pareto front including $\Theta\left(n2^{\frac{n}{2}}\right)$ Pareto vectors (see Fig. 2)

2.1.3. The SCTOP problem

A simple continuous two-objective optimization problem (SCTOP) is described as:

$$\max\ G(x_1, x_2) = (g_1(x_1, x_2), g_2(x_1, x_2)) = (\lceil x_1/\delta \rceil x_2, \lceil x_1/\delta \rceil (1 - x_2)), \tag{2}$$

where $\delta \in (0, 1), x_1, x_2 \in [0, 1]$. Then, the Pareto front is a line segment defined by (see Fig. 3)

$$g_1 + g_2 = \lceil 1/\delta \rceil,\ g_1 \in [0, 1].$$

2.2. Definitions on finite approximations of Pareto fronts

It is ideal to obtain all the Pareto vectors using MOEAs, but this is sometimes impractical, especially when the number of Pareto vectors is extremely large. In this case, a relatively small population is employed to obtain a well-distributed approximation of the Pareto front.

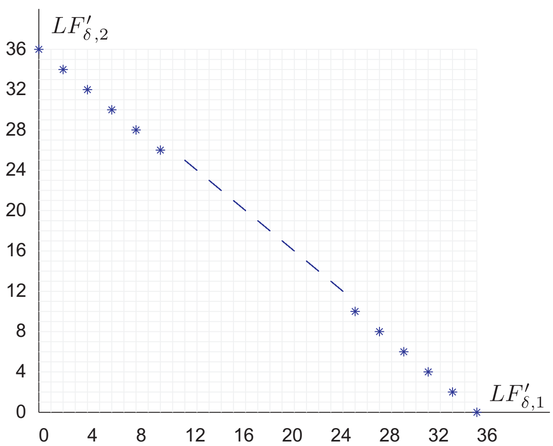

Fig. 2. Objective space of the LF'_δ problem for $\delta = 1$ and $n = 36$.

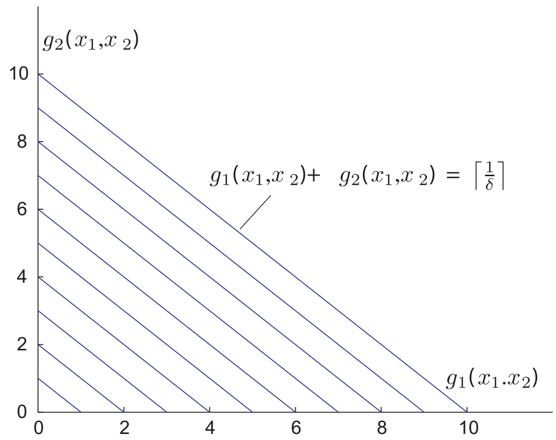

Fig. 3. Objective space of the SCTOP problem for $\delta = 0.1$.

Definition 3. Let $\mathbf{u} = (u_1, \ldots, u_m)$ and $\mathbf{v} = (v_1, \ldots, v_m)$ be two vectors in the objective region S_y of MOP (1).

1. (ϵ-**Dominance**) **u** is said to ϵ-**dominate v** for some $\epsilon > 0$, denoted as $\mathbf{u} \succ_\epsilon \mathbf{v}$, iff for all $i \in \{1, \ldots, m\}$,

   $(1 + \epsilon) \cdot u_i > v_i;$

2. (**Additive** ϵ-**Dominance**) **u** is said to **additively** ϵ-**dominate v** for some $\epsilon > 0$, denoted as $\mathbf{u} \succ_\epsilon^+ \mathbf{v}$, iff for all $i \in \{1, \ldots, m\}$,

   $u_i + \epsilon > v_i.$

Based on the respective definitions of ϵ-dominance and additive ϵ-dominance, we define the following approximations of the Pareto front.

Definition 4. For some $\epsilon > 0$, let F_ϵ^* be a set of Pareto vectors of MOP (1).

1. (ϵ-**Pareto Front**) If any objective vector **v** of MOP (1) is ϵ-dominated by at least one vector $\mathbf{u} \in F_\epsilon^*, F_\epsilon^*$ is called an ϵ-**Pareto front** of MOP (1);

2. (**Additive** ϵ-**Pareto Front**) If any objective vector **v** of MOP (1) is additively ϵ-dominated by at least one vector $\mathbf{u} \in F_\epsilon^*, F_\epsilon^*$ is called an **additive** ϵ-**Pareto front** of MOP (1).

The ϵ-Pareto front[1] is a popular definition of approximate Pareto fronts in most recent theoretical results of MOEAs [26,16,5,12,18,32,19,20]. However, to obtain an ϵ-Pareto front of an MOP for a given diversity index ϵ, the population size μ must be greater than a problem-dependent threshold value [5,18,19]. Otherwise, the distance between two adjacent solutions could be too large to ϵ-dominate some Pareto solutions. Additionally, the population size of a practical MOEA cannot be too large because large populations will lead to high time-complexity of MOEAs. Thus, values of μ and ϵ must be chosen carefully, which could be a difficult undertaking because the MOPs under investigation are usually unfamiliar before they have been studied in depth. Thus, we consider another definition of the finite approximations of Pareto fronts [8], called the adaptive Pareto front.

Definition 5. Let **u** and **v** represent two objective vectors in S_y.

1. (**Weak** δ-**Ball Dominance**) $\forall\ \delta \in \mathbb{R}^+$, **u** is said to **weakly** δ-**ball dominate v** (in short $\mathbf{u} \succeq {}_\delta\mathbf{v}$) with respect to MOP (1), if there exists a $\mathbf{w} \in U(\mathbf{u}, \delta)$ with $\mathbf{w} \succeq \mathbf{v}$, where $U(\mathbf{u}, \delta) = \{\mathbf{w} \in \mathbb{R}^m;\ \|\mathbf{w} - \mathbf{u}\|_2 \leqslant \delta\}$.
2. (**Adaptive Pareto Front**) Let $\mathcal{Q}_A$ be a set of N Pareto vectors, and $\mathbf{u}^* \in \mathcal{Q}_A$ be a Pareto vector with

[1] Although there are some differences between the ϵ-Pareto front and the additive ϵ-Pareto front, they are both defined based on a predetermined index ϵ, which leads to their common shortcoming that ϵ is critical but sometimes hard to predetermine beforehand. Thus, in the following we will not distinguish between them, and will call both the "ϵ-Pareto front".

$$\delta_{\mathbf{u}^*} = \min_{\substack{\mathbf{u} \neq \mathbf{v} \\ \mathbf{u}, \mathbf{v} \in \mathcal{Q}_A}} \|\mathbf{u} - \mathbf{v}\|_2.$$

$\mathcal{Q}_A$ is said to be an **adaptive Pareto front of MOP (1) of size** N, if for any $\mathbf{v} \in S_y$, there exists a $\mathbf{u} \in \mathcal{Q}_A$ satisfying $\mathbf{u} \succeq_{\delta_{\mathbf{u}^*}} \mathbf{v}$. A set $\mathcal{P}_A$ of N feasible solutions is called an **adaptive Pareto set of MOP (1) of size** N, if $F(\mathcal{P}_A)$ is an adaptive Pareto front of MOP (1) of size N.

3. (ϵ-**Adaptive Pareto Front**) Let $\mathcal{Q}_A$ be an adaptive Pareto front of MOP (1) of size N, and let $\mathcal{Q}_A^\epsilon \subset S_y$ be a set of N objective vectors. $\mathcal{Q}_A^\epsilon$ is said to be an ϵ-**adaptive Pareto front of MOP (1) of size** N, if for all $\mathbf{u}^* \in \mathcal{Q}_A$, there exists a $\mathbf{u} \in \mathcal{Q}_A^\epsilon$ such that $\|\mathbf{u}^* - \mathbf{u}\|_2 < \epsilon$. A set $\mathcal{P}_A^\epsilon$ of N feasible solutions is called an ϵ-**adaptive Pareto set of MOP (1) of size** N, if $F(\mathcal{P}_A^\epsilon)$ is an ϵ-adaptive Pareto front of MOP (1) of size N.

Because the adaptive Pareto front of an MOP of size N is confirmed by the population size rather than a given diversity index, it always exists no matter how large N is. Furthermore, when the exact adaptive Pareto front cannot be obtained practically,[2] an ϵ-adaptive Pareto front of an MOP of size N is also acceptable. In the remainder of this paper, the adaptive Pareto front of an MOP of size N and the ϵ-adaptive Pareto front of an MOP of size N are shortened to "the adaptive Pareto front" and "ϵ-adaptive Pareto front", respectively.

2.3. The proposed MOEA

To estimate the runtime of an MOEA, its convergence is usually investigated using a properly defined fitness function satisfying

$$F(\mathbf{x}) \succ F(\mathbf{y}) \Rightarrow d(\mathbf{x}) < d(\mathbf{y}).$$

However, if the Pareto solutions do not have the same fitness value, such a fitness function always drives the individuals to converge to a local part of the Pareto front, which does not meet the requirement of achieving a uniform approximation of the true Pareto front. To overcome this weakness, this paper investigates fitness functions that are compatible with the dominance relation.

Definition 6. Let $d(\mathbf{x})$ be a function defined in the feasible region S_x of MOP (1). $d(\mathbf{x})$ is called a fitness function **compatible with the dominance relation**, if the following hold:

1. $F(\mathbf{x}) \succ F(\mathbf{y}) \Rightarrow d(\mathbf{x}) > d(\mathbf{y})$;
2. $d(\mathbf{x}) = M \iff \mathbf{x} \in PS$, where $M \in \mathbb{R}^+$.

Update strategies based on fitness functions compatible with the dominance relation will not prefer any local part of the Pareto front. Subsequently, by employing extra diversity strategies, MOEAs can obtain a uniform distribution of the Pareto front. A universal framework of $(\mu + 1)$ MOEAs is illustrated by Algorithm 1. First, a population of size μ is generated randomly. Then, a new candidate solution $\mathbf{x}'$ is generated to update the population $\mathcal{P}^{(t)}$ repeatedly until the stopping criterion is satisfied. In this paper, the stopping criterion is finding an adaptive (or ϵ-adaptive) Pareto front.

For discrete MOPs, the "*DGenerate ()*" function described in Algorithm 2 is utilized to generate a new solution, whereas the real-coded $(\mu + 1)$ MOEA employs the update function "*CGenerate ()*" described by Algorithm 3 to solve continuous MOPs. After a new candidate is generated, the "Update" function renews the population via Algorithm 4, where the **governance** relation is used to compare non-dominated solutions.

Definition 7. Let $\mathbf{x}$ and $\mathbf{y}$ be two feasible solutions of MOP (1). It is said that $\mathbf{x}$ **governs** $\mathbf{y}$, denoted as $\mathbf{x} \triangleleft \mathbf{y}$, if it holds that

1. $d(\mathbf{x}) \geqslant d(\mathbf{y})$;
2. $Dist(\mathbf{x}, \mathcal{P}^{(t)} \setminus \{\mathbf{x}, \mathbf{y}\}) > Dist(\mathbf{y}, \mathcal{P}^{(t)} \setminus \{\mathbf{x}, \mathbf{y}\})$,

where $d(\mathbf{x})$ is a fitness function compatible with the dominance relation. The distance function $Dist(\cdot, \cdot)$ is defined by

$$Dist(\mathbf{x}, \mathbf{Q}) = \min_{\mathbf{z} \in \mathbf{Q}} dist(\mathbf{x}, \mathbf{z}) = \min_{\mathbf{z} \in \mathbf{Q}} \|F(\mathbf{x}) - F(\mathbf{z})\|_2.$$

[2] When the investigated MOP is a continuous problem, or when the expected runtime of obtaining the exact adaptive Pareto front is unacceptable, it is impractical to obtain the exact adaptive Pareto front.

Above all, the “Update ()” function tries to save all non-dominated solutions that are found. If the number of non-dominated solutions is less than or equal to the population size μ, they are all saved. If the μ individuals in $\mathcal{P}^{(t)}$ and $\mathbf{x}'$ are non-dominated with each other, the “Update ()” function sorts the population according to the distance value $D(\mathbf{x}, \mathcal{P} \setminus \{\mathbf{x}\})$. An individual $\mathbf{y}$ with $D(\mathbf{y}, \mathcal{P} \setminus \{\mathbf{y}\}) = \min_{\mathbf{x}\in\mathcal{P}} D(\mathbf{x}, \mathcal{P} \setminus \{\mathbf{x}\})$ is called the *worst* individual in the population, denoted as $\mathbf{x}_w$; an individual $\mathbf{y}$ with $D(\mathbf{y}, \mathcal{P} \setminus \{\mathbf{y}\}) = \max_{\mathbf{x}\in\mathcal{P}} D(\mathbf{x}, \mathcal{P} \setminus \{\mathbf{x}\})$ is called the *best* individual in the population, denoted as $\mathbf{x}_b$. If $\mathbf{x}' \lhd \mathbf{x}_w$, then $\mathbf{x}_w \in \mathcal{P}^{(t)}$ is replaced by $\mathbf{x}'$; otherwise, $\mathbf{x}'$ is compared with a randomly selected individual $\mathbf{y}$ and replaces $\mathbf{y}$ if $\mathbf{x}' \lhd \mathbf{y}$. Because only one individual is generated at each generation, in this paper, we perform the runtime analysis by estimating expected iterations of the $(\mu + 1)$ MOEA consisting of Algorithms 1, 2 or 3 and 4, and “runtime” refers to the number of iterations before stopping.

Algorithm 1. Multi-objective Evolutionary Algorithm (MOEA)

Set generation $t = 1$;
Randomly generate a population $\mathcal{P}^{(t)}$ of μ individuals;
while the stop criterion is not satisfied
    $\mathbf{x}' = Generate(\mathcal{P}^{(t)})$;
    $\mathcal{P}^{(t+1)} = Update(\mathcal{P}^{(t)}, \mathbf{x}')$;
    $t = t + 1$;
end while
Output the results.

Algorithm 2. $DGenerate(\mathcal{P}^{(t)})$

Select an individual $\mathbf{x}$ from $\mathcal{P}^{(t)}$ randomly;
Generate a candidate $\mathbf{x}'$ by flipping each bit of $\mathbf{x}$ with probability $\frac{1}{n}$;
Output $\mathbf{x}'$.

Algorithm 3. $CGenerate(\mathcal{P}^{(t)})$

Select an individual $\mathbf{x}$ from $\mathcal{P}^{(t)}$ randomly;
Generate a candidate $\mathbf{x}'$ by

$$\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x},$$

where $\Delta\mathbf{x}$ is a random variable obeying the normal distribution $N(\mathbf{0}, \boldsymbol{\sigma})$.
Output $\mathbf{x}'$.

Algorithm 4. $Update(\mathcal{P}^{(t)}, \mathbf{x}')$

if $\exists\mathbf{x} \in \mathcal{P}^{(t)}$ such that $\mathbf{x} \succ \mathbf{x}' \vee F(\mathbf{x}) = F(\mathbf{x}')$ **then**
    $\mathcal{P} = \mathcal{P}^{(t)}$;
else
    $\mathcal{P} = \mathcal{P}^{(t)} \setminus \{\mathbf{x} \in \mathcal{P}^{(t)} | \mathbf{x}' \succ \mathbf{x}\}$.
    if $Size(\mathcal{P}) \leqslant \mu - 1$ **then**
        $\mathcal{P} = \mathcal{P} \cup \{\mathbf{x}'\}$;
    else
        Sort $\mathcal{P}$ according to the distance value $D(\mathbf{x}, \mathcal{P} \setminus \{\mathbf{x}\})$;
        if $\mathbf{x}' \lhd \mathbf{x}_w$ **then**
            $\mathcal{P} = \mathcal{P} \setminus \{\mathbf{x}_w\} \cup \{\mathbf{x}'\}$;
        else

```
            Select an individual y from P randomly;
            if x' ◁ y then
                P = P \ {y} ∪ {x'};
            end if
        end if
    end if
end if
Output P.
```

Based on a fitness function $f(\mathbf{x})$ compatible with the dominance relation, the feasible region is a totally ordered set. Thus, the selection pressure of update strategies based on $f(\mathbf{x})$ could be greater than update strategies based only on the dominance relation, which make MOEAs efficient for many-objective optimization problems. A fitness function compatible with the dominance relation has been presented in [8], where the true Pareto front of the investigated MOPs must be known. However, if the Pareto front is not known in advance, an MOEA can employ a practical approximation defined as follows.

Example 1. Let $\mathcal{P}$ be the population of the $(\mu+1)$ MOEA described in Algorithms 1–4, and let $\mathbf{x}_1,\ldots,\mathbf{x}_\mu$ be μ non-dominated individuals in $\mathcal{P}$. When the $(\mu+1)$ MOEA is utilized to solve MOP (1), a new candidate $\mathbf{x}'$ competes with $\mathbf{x}$ to survive into the next generation. Let $\mathbf{x}_{k_1},\ldots,\mathbf{x}_{k_m}$ be the m nearest non-dominated individuals of $\mathbf{x}$ when considered in terms of Euclidean distances in the objective space. Then, we can define the fitness values of $\mathbf{x}$ and $\mathbf{x}'$ by

$$d_A(\mathbf{y}) = \sum_{i=1}^{m}\left(f_i(\mathbf{y}) - \frac{1}{m}\sum_{j=1}^{m} f_i(\mathbf{x}_{k_j})\right). \tag{3}$$

Although the fitness function $d_A(\cdot)$ is defined via the present population of an MOEA, it actually holds that $d_A(\mathbf{x}') < d_A(\mathbf{x})$ if the new candidate $\mathbf{x}' \succ \mathbf{x}$. According to the definition, if the $(\mu+1)$ MOEA has obtained μ Pareto solutions and the diversity of the population is well preserved, the fitness value of Pareto solutions in the population will be almost identical. In this case, the difference among the fitness values of all Pareto solutions could be small enough if the population size μ is sufficiently large.

3. Runtime analysis of the $(\mu+1)$ MOEA on three investigated MOPs

3.1. Obtaining the whole Pareto front of LOTZ

For LOTZ, Laumanns et. al [27,28] showed that the SEMO with an unbounded archive can obtain the whole Pareto front in expected runtime $\Theta(n^3)$. Moreover, when the population size is fixed to be μ greater than $n+1$ (the total number of Pareto vectors of LOTZ), the $(\mu+1)$ simple indicator-based evolutionary algorithm (SIBEA) can locate the $n+1$ Pareto vectors of LOTZ in $\mathcal{O}(\mu n^2)$ [5]. Defining $d(\mathbf{x})$ to be the sum of two objective values of an individual $\mathbf{x}$, we obtain the following result for the expected runtime of the $(\mu+1)$ MOEA with $\mu \leqslant n+1$.

Theorem 1. *When $\mu \leqslant n+1$, the $(\mu+1)$ MOEA consisting of Algorithms 1, 2 and 4 achieves an adaptive-Pareto front of the LOTZ problem of size μ in the expected runtime $\mathcal{O}(\mu n^2 + \mu^2 n^{\lfloor \frac{n}{\mu-1} \rfloor})$.*

Proof. Denote $\mathcal{P}$ as the population of the $(\mu+1)$ MOEA, and $\mathbf{x}_w$ as the worst individual in the population. The evolving process of the $(\mu+1)$ MOEA contains two stages:

1. **Converging to the Pareto front.**
   In the first stage, the population attempts to find μ Pareto solutions. Because the probability of generating a new candidate $\mathbf{x}'$ that dominates $\mathbf{x}$ is $\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}$, $\mathbf{x}'$ will replace $\mathbf{x}$ in the expected runtime $\mathcal{O}(n)$. For at most n steps, a Pareto solution is obtained in the expected runtime $\mathcal{O}(n^2)$.
   When k Pareto solutions have been obtained, another Pareto solution can be generated with a probability greater than $\frac{1}{k}\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}$. Then, one more Pareto solution can enter the population in the expected runtime $\mathcal{O}(kn)$. Thus, to obtain μ Pareto solutions, the expected runtime is $\mathcal{O}(\mu^2 n)$.
   Subsequently, the $(\mu+1)$ MOEA can obtain μ Pareto solutions of LOTZ in the expected runtime $\mathcal{O}(n^2+\mu^2 n)$.
2. **Spreading along the Pareto front.**
   After the population $\mathcal{P}$ has changed into a set of Pareto solutions, the diversity strategy drives the population to evolve into a reasonable representation of the Pareto front.

(a) First, two of the obtained Pareto vectors will move to the two respective boundary positions of the Pareto front. Because the leftmost and the rightmost vectors can always move toward the two boundary points of the Pareto front with a probability greater than $\frac{1}{\mu}\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}$, the total expected time of at most n steps is $\mathcal{O}(\mu n^2)$. Then, if an adaptive Pareto front is not obtained, additional iterations are needed for the $(\mu+1)$ MOEA to obtain an adaptive Pareto front. Meanwhile, if two individuals have reached the two boundary points of the Pareto front, they would not move further, because such a move would lead to a decrease in the value of $D(\mathbf{x}, \mathcal{P} \setminus \{\mathbf{x}\})$. Thus, in the following, we consider only the case when there are two individuals located on the boundary places of the Pareto front.

(b) If $1 < \frac{n}{\mu-1} < 2$, the individuals will move along the Pareto front with a probability greater than $\frac{1}{\mu}\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}$. Because the total number of Pareto solutions is n, an adaptive Pareto front can be obtained after at most n steps. The expected runtime of this procedure is $\mathcal{O}(\mu n^2)$.

(c) If $\frac{n}{\mu-1}$ is equal to an integer $k \geqslant 2$, the spreading process of the population can be divided into two steps.

i. When $\min_{\substack{\mathbf{x}\neq\mathbf{y}\\ \mathbf{x},\mathbf{y}\in\mathcal{P}}} dist(\mathbf{x}, \mathbf{y}) < \lfloor\frac{k}{2}\rfloor$, there are two points $\mathbf{x}^*, \mathbf{y}^*$ with $dist(\mathbf{x}^*, \mathbf{y}^*) \leqslant \lfloor\frac{k}{2}\rfloor - 1$. Moreover, if there exists no $\mathbf{x}_P \in PS \setminus \mathcal{P}$ with $Dist(\mathbf{x}_P, \mathcal{P}) = \lfloor\frac{k}{2}\rfloor$, the distance between any two adjacent vectors in $LOTZ(\mathcal{P})$ is always less than $2\lfloor\frac{k}{2}\rfloor$, which does not hold because $(\mu-2)\times(2\times\lfloor\frac{k}{2}\rfloor-1)+\lfloor\frac{k}{2}\rfloor-1$ is necessarily less than n. Thus, the population can evolve into a set of Pareto solutions with

$$\min_{\substack{\mathbf{x}\neq\mathbf{y}\\ \mathbf{x},\mathbf{y}\in\mathcal{P}}} dist(\mathbf{x}, \mathbf{y}) \geqslant \left\lfloor\frac{k}{2}\right\rfloor,$$

which can be obtained as follows.
If there exists an $\mathbf{x}_P \in PS \setminus \mathcal{P}$ with $Dist(\mathbf{x}_P, \mathcal{P}) = \lfloor\frac{k}{2}\rfloor$, it is generated with a probability greater than $\frac{1}{\mu}\left(\frac{1}{n}\right)^{\lfloor\frac{k}{2}\rfloor}\left(1-\frac{1}{n}\right)^{n-\lfloor\frac{k}{2}\rfloor}$. Then, when there exists an individual $\mathbf{x}_w \in \mathcal{P}$ with $Dist(\mathbf{x}_w, \mathcal{P} \setminus \{\mathbf{x}_w\}) < \lfloor\frac{k}{2}\rfloor$, it will be replaced by $\mathbf{x}_P$ in the expected runtime $\mathcal{O}(\mu n^{\lfloor\frac{k}{2}\rfloor})$. For the worst case,

$$Dist(\mathbf{x}, \mathcal{P} \setminus \{\mathbf{x}_w\}) < \left\lfloor\frac{k}{2}\right\rfloor$$

holds for all $\mathbf{x} \in \mathcal{P}$. Thus, after at most μ updates, the distance between any two individuals in the population is greater than or equal to $\lfloor\frac{k}{2}\rfloor$, and the total expected runtime of this procedure is $\mathcal{O}(\mu^2 n^{\lfloor\frac{k}{2}\rfloor})$.

ii. In the second step, the whole population will evolve into an adaptive Pareto front. Denoting $M = \min_{\substack{\mathbf{x}\neq\mathbf{y}\\ \mathbf{x},\mathbf{y}\in\mathcal{P}}} dist(\mathbf{x}, \mathbf{y})$, we have $M \leqslant k$. Then, if there exists a Pareto solution $\mathbf{x}_P \in PS \setminus \mathcal{P}$ with $Dist(\mathbf{x}_P, \mathcal{P}) = M+1, \mathbf{x}_P$ is generated to replace the worst individual $\mathbf{x}_w$ with $Dist(\mathbf{x}_w, \mathcal{P} \setminus \{\mathbf{x}_w\}) = M$ in expected time $\mathcal{O}(\mu n^{M+1})$, and one of the following two cases arises.

A. For at most μ times, we have $\min_{\mathbf{x}\neq\mathbf{y}} dist(\mathbf{x}, \mathbf{y}) = M+1$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{P}$, and the expected time is $\mathcal{O}(\mu^2 n^{M+1})$.

B. Otherwise, after $t (t \leqslant \mu)$ updates the minimum value of the distances between any two different individuals in the population is still M, and there are no Pareto solutions $\mathbf{x}_P \in PS \setminus \mathcal{P}$ with $Dist(\mathbf{x}_P, \mathcal{P}) \geqslant M$. Thus, an adaptive Pareto front is obtained, and the expected time is also $\mathcal{O}(\mu^2 n^{M+1})$.

After at most $\lceil\frac{k}{2}\rceil$ repetitions of the aforementioned procedure, the adaptive Pareto front is obtained in the expected runtime

$$\mathcal{O}(\mu^2 n^{M+1}) + \mathcal{O}(\mu^2 n^{M+2}) + \ldots + \mathcal{O}(\mu^2 n^k) = \mathcal{O}(\mu^2 n^k).$$

(d) If $k < \frac{n}{\mu-1} < k+1, k = 2, 3, \ldots$, from the arguments for the case of $\frac{n}{\mu-1}$ we can conclude that after $\mathcal{O}(\mu^2 n^k)$ expected steps, the population will evolve to a set of Pareto solutions in which the distance between two adjacent points is less than or equal to k. Then, if $\mathcal{P}$ is not an adaptive Pareto set, the individuals will move along the Pareto front with a probability greater than $\frac{1}{\mu}\frac{1}{n}\left(1-\frac{1}{n}\right)^{n-1}$. Because the total number of Pareto solutions is n, an adaptive Pareto front can be obtained after at most n steps, and the expected runtime of this procedure is $\mathcal{O}(\mu n^2)$. Consequently, the expected runtime in this case is also $\mathcal{O}(\mu^2 n^k)$.

In conclusion, the expected runtime until the $(\mu+1)$ MOEA has obtained an adaptive Pareto front of LOTZ of size μ is $\mathcal{O}(\mu n^2 + \mu^2 n^{\lfloor\frac{n}{\mu-1}\rfloor})$. □

Because the $(\mu+1)$ MOEA obtains $\mu(\leqslant n+1)$ Pareto vectors in $\mathcal{O}(n^2+\mu^2 n)$, by setting $\mu=n+1$ it accommodates the whole Pareto front of LOTZ in expected runtime $\mathcal{O}(n^3)$. When $\mu<n+1$, to obtain an adaptive Pareto front extra iterations are required to create a uniform distribution along the Pareto front, and accordingly, the expected runtime of this process is $\mathcal{O}(\mu^2 n^{\left\lfloor\frac{n}{\mu-1}\right\rfloor})$. Thus, when $\frac{n}{2}+1<\mu\leqslant n+1$, it holds that $\left\lfloor\frac{n}{\mu-1}\right\rfloor=1$, and both the expected runtime of the converging process and that of the spreading process are $\mathcal{O}(n^3)$. However, when $\mu<\frac{n}{2}+1$, $\left\lfloor\frac{n}{\mu-1}\right\rfloor$ is necessarily greater than or equal to 2, and the expected runtime of the spreading process is at least $\mathcal{O}(\mu^2n^2)$, which is greater than that of the converging process. Then, regarding the LOTZ problem, we can conclude that for the proposed $(\mu+1)$ MOEA, it is better to obtain all of the Pareto vectors instead of a uniformly distributed representation of the Pareto front, even if the problem size n is very large.

3.2. Achieving a reasonable approximate Pareto front of the LF'_δ problem

It has been shown that when the population size is greater than some problem-dependent value, some MOEAs can obtain an ϵ-approximation of an exponentially large Pareto front with $\epsilon=\delta$ in the polynomial expected runtime [5,18,19]. By defining $d(\mathbf{x})$ to be the sum of two objective values of an individual $\mathbf{x}$, we estimate here the expected runtime until the $(\mu+1)$ MOEA has achieved a reasonable approximation of the Pareto front for the case $\mu\leqslant\frac{n}{2}+1$.

Theorem 2. *When $\mu\leqslant\frac{n}{2}+1$, the $(\mu+1)$ MOEA consisting of Algorithms 1, 2 and 4 achieves a δ-adaptive Pareto front of LF'_δ of size μ in the expected runtime $\mathcal{O}\left(\mu n\log n+\mu n^{\left\lfloor\frac{n/2}{\mu-1}\right\rfloor}\log n\right)$.*

Proof. Let $\mathbf{x}$ and $\mathbf{y}$ be two individuals in the population $\mathcal{P}$. If $|\ell(\mathbf{x})|_1=|\ell(\mathbf{y})|_1, dist(\mathbf{x},\mathbf{y})<\delta$ holds. Otherwise, we have

$$(2k+1)\delta>dist(\mathbf{x},\mathbf{y})>(2k-1)\delta \tag{4}$$

when $||\ell(\mathbf{x})|_1-|\ell(\mathbf{y})|_1|=k,\ k=1,2,\ldots,\frac{n}{2}$.

1. First, two obtained Pareto vectors can move to the two boundary points of the Pareto front. If there are i 1-bits in the first half part of the present individual $\mathbf{x}$, the probability of generating a new candidate $\mathbf{y}$ with $||\ell(\mathbf{x})|_1-|\ell(\mathbf{y})|_1|=1$ is greater than

$$\min\left\{\frac{n}{2}-i,i\right\}\frac{1}{\mu}\frac{1}{n}\left(1-\frac{1}{n}\right)^{\frac{n}{2}-1}.$$

Thus, it costs at most $\frac{n}{2}$ steps for two individuals to move to two boundary positions of the Pareto front, and the expected runtime of this procedure is less than

$$\sum_{i=0}^{\frac{n}{2}}\frac{\mu n}{\min\left\{\frac{n}{2}-i,i\right\}\left(1-\frac{1}{n}\right)^{\frac{n}{2}-1}}=\mathcal{O}(\mu n\log n).$$

2. If $1\leqslant\frac{n/2}{\mu-1}<2$, the individuals will move along the Pareto front with a probability greater than

$$\min\left\{\frac{n}{2}-i,i\right\}\frac{1}{\mu}\frac{1}{n}\left(1-\frac{1}{n}\right)^{\frac{n}{2}-1},$$

where i is the number of 1-bits in the first half part of the individual. Because the total number of Pareto solutions is $\frac{n}{2}$, after at most n steps a δ-adaptive Pareto front can be obtained, and the expected runtime of this procedure is also $\mathcal{O}(\mu n\log n)$.

3. If $\frac{n/2}{\mu-1}$ is an integer $k\geqslant 2$, the spreading process of the population is divided into two steps.
   (a) When

$$\min_{\substack{\mathbf{x}\neq\mathbf{y}\\ \mathbf{x},\mathbf{y}\in\mathcal{P}}}||\ell(\mathbf{x})|_1-|\ell(\mathbf{y})|_1|<\left\lfloor\frac{k}{2}\right\rfloor,$$

there are two points $\mathbf{x}^*,\mathbf{y}^*$ with $||\ell(\mathbf{x}^*)|_1-|\ell(\mathbf{y}^*)|_1|\leqslant\lfloor\frac{k}{2}\rfloor-1$. If there are no $\mathbf{x}_P\in PS\setminus\mathcal{P}$ with $\min_{\mathbf{x}\in\mathcal{P}}||\ell(\mathbf{x}_P)|_1-|\ell(\mathbf{x})|_1|=\lfloor\frac{k}{2}\rfloor$, the distance between any two adjacent vectors in $LF'_\delta(\mathcal{P})$ is always less than $2\lfloor\frac{k}{2}\rfloor$, which does not hold because $(\mu-2)\times(2\times\lfloor\frac{k}{2}\rfloor-1)+\lfloor\frac{k}{2}\rfloor-1$ is necessarily less than $n/2$. Thus, the population can evolve into a set of Pareto solutions with

$$\min_{\substack{\mathbf{x}\neq\mathbf{y}\\ \mathbf{x},\mathbf{y}\in\mathcal{P}}}||\ell(\mathbf{x})|_1-|\ell(\mathbf{y})|_1|\geqslant\left\lfloor\frac{k}{2}\right\rfloor,$$

which can be obtained as follows.
If there exists an $\mathbf{x}_P \in PS \setminus \mathcal{P}$ with $\min_{\mathbf{x}\in\mathcal{P}} ||\ell(\mathbf{x}_P)|_1 - |\ell(\mathbf{x})|_1| = \lfloor \frac{k}{2} \rfloor$, it is generated with a probability greater than

$$\frac{1}{\mu}\min\left\{\binom{|\ell(\mathbf{x})|_1}{k},\binom{\frac{n}{2}-|\ell(\mathbf{x})|_1}{k}\right\}\left(\frac{1}{n}\right)^{\lfloor\frac{k}{2}\rfloor}\left(1-\frac{1}{n}\right)^{n-\lfloor\frac{k}{2}\rfloor},$$

where i represents the number of 1-bits in the first half part of the selected individual, and $\mathbf{x}_P$ will replace the worst individual $\mathbf{x}_w \in \mathcal{P}$ in expected time

$$\frac{\mu n^{\lfloor\frac{k}{2}\rfloor}}{\min\left\{\binom{|\ell(\mathbf{x})|_1}{k},\binom{\frac{n}{2}-|\ell(\mathbf{x})|_1}{k}\right\}} \leqslant \frac{\mu n^{\lfloor\frac{k}{2}\rfloor}}{\min\left\{|\ell(\mathbf{x})|_1,\frac{n}{2}-|\ell(\mathbf{x})|_1\right\}}.$$

After at most μ updates, the distance between any two individuals is greater than or equal to $\lfloor\frac{k}{2}\rfloor$. Let the μ generated candidates be $\mathbf{x}_1,\ldots,\mathbf{x}_\mu$. The total expected runtime of this procedure is less than

$$\sum_{j=1}^{\mu}\frac{n^{\lfloor\frac{k}{2}\rfloor}}{\min\left\{|\ell(\mathbf{x}_j)|_1,\frac{n}{2}-|\ell(\mathbf{x}_j)|_1\right\}} \leqslant \sum_{i=1}^{n}\frac{n^{\lfloor\frac{k}{2}\rfloor}}{\min\left\{i,\frac{n}{2}-i\right\}} = \mathcal{O}\left(\mu n^{\lfloor\frac{k}{2}\rfloor}\log n\right).$$

(b) In the second step, a δ-adaptive Pareto front can be obtained. Denoting $M = \min_{\mathbf{x},\mathbf{y}\in\mathcal{P}} ||\ell(\mathbf{x})|_1 - |\ell(\mathbf{y})|_1|$, we have $M \leqslant k$. Then, if there exists a Pareto vector $\mathbf{x}_P \notin \mathcal{P}$ with $||\ell(\mathbf{x})|_1 - |\ell(\mathbf{y})|_1| = M+1$, it is generated in expected time $\frac{\mu n^{M+1}}{\min\left\{|\ell(\mathbf{x})|_1,\frac{n}{2}-|\ell(\mathbf{x})|_1\right\}}$. Thus, after at most μ updates, the total expected time is $\mathcal{O}(\mu n^{M+1}\log n)$, and the minimum distance is greater than or equal to $M+1$. If this process is repeated when possible, then after at most $\lceil\frac{k}{2}\rceil$ times, we will obtain a δ-adaptive Pareto front of LF'_δ of size μ in the expected runtime

$$\mathcal{O}(\mu n^{M+1}\log n) + \mathcal{O}(\mu n^{M+2}\log n) + \ldots + \mathcal{O}(\mu n^{k}\log n) = \mathcal{O}(\mu n^{k}\log n).$$

4. If $k < \frac{n/2}{\mu-1} < k+1, k = 2,3,\ldots$, the expected runtime is $\mathcal{O}(\mu n^k \log n)$, which can be obtained by arguments similar to those made in case 3) in proof of Theorem 1.

In conclusion, the $(\mu+1)$ MOEA achieves a δ-adaptive Pareto front of LF'_δ of size μ in expected time $\mathcal{O}(\mu n\log n + \mu n^{\lfloor\frac{n/2}{\mu-1}\rfloor}\log n)$. □

The result in Theorem 2 is similar to that of Theorem 1 because the Pareto front of LF'_δ can be divided into $\frac{n}{2}+1$ grids according to the number of 1-bits in the first half part of Pareto solutions. When $\mu = \frac{n}{2}+1$, in expected runtime $\mathcal{O}(\mu n\log n)$ the $(\mu+1)$ can obtain a δ-adaptive Pareto front of size μ, which is also a δ-approximation of the Pareto front. However, when $\mu < \frac{n}{2}+1$, we can conclude that the expected runtime of the $(\mu+1)$ MOEA for the LF'_δ problem is actually exponential. The reason is that when n is increased, the total number of grids on the Pareto front is also increased. Then, if no adaptive mutation strategies are employed, the expected runtime of exploring the whole Pareto front will rise in the order of $\mathcal{O}(\mu n^{\left\lfloor\frac{n/2}{\mu-1}\right\rfloor}\log n)$. However, the expected polynomial runtime can also be obtained if μ is kept to $\Theta(n)$. That is, when n increases, we also enlarge the population size μ to keep $k \leqslant \frac{n/2}{\mu-1} < k+1$ for a constant integer k. In this way, the total expected runtime is $\mathcal{O}(n^{k+1}\log n)$, and the space complexity of the $(\mu+1)$ MOEA increases on the order of $\Theta(n)$.

3.3. Solving the SCTOP in the expected polynomial runtime

In this section, we investigate the runtime of $(\mu+1)$ MOEA for the SCTOP problem. Although the techniques employed here are similar to those utilized for the DMOPs, we come to an entirely different result for the expected runtime of $(\mu+1)$ MOEA.

$\forall\ \mathbf{x} = (x_1,x_2) \in [0,1]^2$, the fitness function is defined as $d(\mathbf{x}) = \lceil\frac{x_1}{\delta}\rceil$. Thus, $\mathbf{x}$ is a Pareto solution if and only if $d(\mathbf{x}) = \lceil\frac{1}{\delta}\rceil$.

Theorem 3. *For SCTOP, the $(\mu+1)$ MOEA consisting of Algorithms 1, 3 and 4 can find the first Pareto solution in the expected runtime $\mathcal{O}(\sigma^2)$.*

Proof. According to the update strategy of the $(\mu+1)$ MOEA, the individual $\mathbf{x} = (x_1,x_2)$ will be replaced if a dominating candidate $\mathbf{y} = (y_1,y_2)$ is generated. Following such a strategy, the expected improvement of $d(\mathbf{x})$ is

$$\begin{aligned}\mathbf{E}\left[d(\mathbf{y}) - d(\mathbf{x})|d(\mathbf{x}) = \left\lceil\frac{x_1}{\delta}\right\rceil\right] &= \mathbf{E}\left[\left\lceil\frac{y_1}{\delta}\right\rceil - \left\lceil\frac{x_1}{\delta}\right\rceil\middle| d(\mathbf{x}) = \left\lceil\frac{x_1}{\delta}\right\rceil\right]\\ &= \iint_D\left(\left\lceil\frac{y_1}{\delta}\right\rceil - \left\lceil\frac{x_1}{\delta}\right\rceil\right)\frac{1}{2\pi\sigma^2}\exp\left\{-\frac{(y_1-x_1)^2+(y_2-x_2)^2}{2\sigma^2}\right\}dy_1dy_2,\end{aligned}$$

where $D = \{(y_1, y_2) | \lceil\frac{y_1}{\delta}\rceil y_2 \geqslant \lceil\frac{x_1}{\delta}\rceil x_2,\ (1 - \lceil\frac{y_1}{\delta}\rceil) y_2 \geqslant (1 - \lceil\frac{x_1}{\delta}\rceil) x_2\}$. Thus,

$$\mathbf{E}\Big[d(\mathbf{y}) - d(\mathbf{x}) | d(\mathbf{x}) = \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big] \geqslant \frac{1}{2\pi\sigma^2}\int_{\delta\lceil\frac{x_1}{\delta}\rceil}^{1}\Big(\Big\lceil\frac{y_1}{\delta}\Big\rceil - \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big)\exp\Big\{-\frac{(y_1-x_1)^2}{2\sigma^2}\Big\}dy_1\int_{\frac{\delta x_2}{y_1}\lceil\frac{x_1}{\delta}\rceil}^{1-\frac{\delta(1-x_2)}{y_1}\lceil\frac{x_1}{\delta}\rceil}\exp\Big\{-\frac{(y_2-x_2)^2}{2\sigma^2}\Big\}dy_2$$

By Taylor's theorem, we know that

$$\int_{\frac{\delta x_2}{y_1}\lceil\frac{x_1}{\delta}\rceil}^{1-\frac{\delta(1-x_2)}{y_1}\lceil\frac{x_1}{\delta}\rceil}\exp\Big\{-\frac{(y_2-x_2)^2}{2\sigma^2}\Big\}dy_2 = \exp\left\{-\frac{\left(\frac{\delta x_2}{y_1}\lceil\frac{x_1}{\delta}\rceil - x_2\right)^2}{2\sigma^2}\right\}\Big(1 - \Big\lceil\frac{x_1}{\delta}\Big\rceil\frac{\delta}{y_1}\Big) + o\Big(\Big(1 - \Big\lceil\frac{x_1}{\delta}\Big\rceil\frac{\delta}{y_1}\Big)^2\Big),$$

and then,

$$\begin{aligned}
&\mathbf{E}\Big[d(\mathbf{y}) - d(\mathbf{x}) | d(\mathbf{x}) = \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big]\\
\geqslant\ & \frac{1}{2\pi\sigma^2}\int_{\delta\lceil\frac{x_1}{\delta}\rceil}^{1}\Big(\Big\lceil\frac{y_1}{\delta}\Big\rceil - \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big)\exp\Big\{-\frac{(y_1-x_1)^2}{2\sigma^2}\Big\}dy_1\int_{\frac{\delta x_2}{y_1}\lceil\frac{x_1}{\delta}\rceil}^{1-\frac{\delta(1-x_2)}{y_1}\lceil\frac{x_1}{\delta}\rceil}\exp\Big\{-\frac{(y_2-x_2)^2}{2\sigma^2}\Big\}dy_2\\
=\ & \frac{1}{2\pi\sigma^2}\int_{\delta\lceil\frac{x_1}{\delta}\rceil}^{1}\Big(\Big\lceil\frac{y_1}{\delta}\Big\rceil - \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big)\exp\left\{-\frac{\left(\frac{\delta x_2}{y_1}\lceil\frac{x_1}{\delta}\rceil - x_2\right)^2}{2\sigma^2}\right\}\Big(1 - \Big\lceil\frac{x_1}{\delta}\Big\rceil\frac{\delta}{y_1}\Big)dy_1\\
+\ & \frac{1}{2\pi\sigma^2}\int_{\delta\lceil\frac{x_1}{\delta}\rceil}^{1}\Big(\Big\lceil\frac{y_1}{\delta}\Big\rceil - \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big)\cdot o\Big(\Big(1 - \Big\lceil\frac{x_1}{\delta}\Big\rceil\frac{\delta}{y_1}\Big)^2\Big)dy_1.\\
\geqslant\ & \frac{1}{2\pi\sigma^2}\int_{\delta\lceil\frac{x_1}{\delta}\rceil}^{1}\Big(\frac{y_1}{\delta} - \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big)\exp\left\{-\frac{\left(\frac{\delta x_2}{y_1}\lceil\frac{x_1}{\delta}\rceil - x_2\right)^2}{2\sigma^2}\right\}\Big(1 - \Big\lceil\frac{x_1}{\delta}\Big\rceil\frac{\delta}{y_1}\Big)dy_1\\
+\ & \frac{1}{2\pi\sigma^2}\int_{\delta\lceil\frac{x_1}{\delta}\rceil}^{1}\Big(\frac{y_1}{\delta} - \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big)\cdot o\Big(\Big(1 - \Big\lceil\frac{x_1}{\delta}\Big\rceil\frac{\delta}{y_1}\Big)^2\Big)dy_1.
\end{aligned}$$

Moreover, based on the fact that

$$\int_{\delta\lceil\frac{x_1}{\delta}\rceil}^{1}\Big(\frac{y_1}{\delta} - \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big)\exp\left\{-\frac{\left(\frac{\delta x_2}{y_1}\lceil\frac{x_1}{\delta}\rceil - x_2\right)^2}{2\sigma^2}\right\}\Big(1 - \Big\lceil\frac{x_1}{\delta}\Big\rceil\frac{\delta}{y_1}\Big)dy_1 = \frac{1}{3}\frac{1}{\delta^2\lceil\frac{x_1}{\delta}\rceil}\Big(1 - \delta\Big\lceil\frac{x_1}{\delta}\Big\rceil\Big)^3 + o\Big(\Big(1 - \delta\Big\lceil\frac{x_1}{\delta}\Big\rceil\Big)^3\Big),$$

we come to the result that $\mathbf{E}\big[d(\mathbf{y}) - d(\mathbf{x}) | d(\mathbf{x}) = \lceil\frac{x_1}{\delta}\rceil\big]$ is $\Omega\Big(\frac{1}{\sigma^2\delta x_1}\Big)$. Thus, there exists a constant $k \in \mathbb{R}$ such that

$$\mathbf{E}\Big[d(\mathbf{y}) - d(\mathbf{x}) | d(\mathbf{x}) = \Big\lceil\frac{x_1}{\delta}\Big\rceil\Big] \geqslant k\frac{1}{\sigma^2\delta x_1} \geqslant k\frac{1}{\sigma^2\delta}.$$

By Theorem 1 in [17], we know that the expected runtime to find a solution $\mathbf{x}$ with $d(\mathbf{x}) = \lceil\frac{1}{\delta}\rceil$ is $\mathcal{O}(\sigma^2)$. □

$\forall\ \varepsilon > 0$, the Pareto front of SCTOP can be divided into $N = \Big\lceil\frac{\sqrt{2}}{\varepsilon}\Big\rceil$ grids, where $\sqrt{2}$ is the length of the Pareto front. From the leftmost grid to the rightmost grid, number the grids as $1, 2, \ldots, \lceil\frac{\sqrt{2}}{\varepsilon}\rceil$, and set $[\mathbf{x}] = i$ when $G(\mathbf{x})$ is located in the i^{th} grid. $\forall\ \mathbf{x}, \mathbf{y} \in PS$, the grid distance between two individuals is defined as

$$gdist(\mathbf{x}, \mathbf{y}) = |[\mathbf{x}] - [\mathbf{y}]|,$$

and the grid distance between an individual $\mathbf{x}$ and a set of individuals $\mathbf{Q}$ is defined as

$$Gdist(\mathbf{x}, \mathbf{Q}) = \min_{\mathbf{y}\in\mathbf{Q}} gdist(\mathbf{x}, \mathbf{y}).$$

Then, if two individuals $\mathbf{x}$ and $\mathbf{y}$ are located with a grid distance m, it holds that

$$(m-1)\varepsilon \leqslant dist(\mathbf{x}, \mathbf{y}) \leqslant (m+1)\varepsilon.$$

Denote $\mathbf{x}'$ to be the new candidate solution generated by a mutation on $\mathbf{x}$. Then, $\forall m \in \mathbb{Z}^+$, the infimum of the probability that $\mathbf{y}$ is generated with $gdist(\mathbf{x}, \mathbf{y}) = m$ is

$$P_m(\varepsilon) \geqslant \frac{1}{2\pi\sigma^2}\int_{(\lceil 1/\varepsilon\rceil - 1)\varepsilon}^{1}dy_1\int_{m\varepsilon/\sqrt{2}}^{(m+1)\varepsilon/\sqrt{2}}\exp\Big\{-\frac{(y_1-x_1)^2 + y_2^2}{2\sigma^2}\Big\}dy_2, \tag{5}$$

and the following theorem holds.

Theorem 4. *$\forall \varepsilon > 0$, in the expected runtime $\mathcal{O}(\frac{\sigma^2\mu^2}{\varepsilon}e^{\frac{2}{\sigma^2\mu^2}})$, the $(\mu+1)$ MOEA consisting of Algorithms 1, 3 and 4 obtains an ε-adaptive Pareto front of SCTOP when $\mu < \frac{\lceil\frac{2\sqrt{2}}{\varepsilon}\rceil - 1}{3} + 1$.*

Proof. when $\mu < \frac{\lceil\frac{2\sqrt{2}}{\varepsilon}\rceil - 1}{3} + 1, \frac{N-1}{\mu-1} \geqslant 4$. The evolving process of the population is as follows.

1. If $\frac{N-1}{\mu-1}$ is equal to an integer $k \geqslant 4$, the spreading process of the population can be divided into two procedures.
   (a) Denote $\mathbf{x}^* = \arg\min_{\mathbf{x}\in\mathcal{P}}[\mathbf{x}]$ and $\mathbf{y}^* = \arg\max_{\mathbf{x}\in\mathcal{P}}[\mathbf{x}]$. Then the images of $\mathbf{x}^*$ and $\mathbf{y}^*$ under the map G can move to the two boundary grids of the Pareto front, respectively. Because a Pareto vector of SCTOP can jump from one grid to an adjacent grid with a probability greater than $P_1(\varepsilon)$, this process will last for $\mathcal{O}(\frac{\mu}{P_1(\varepsilon)})$ iterations at expectation. For at most N steps, $G(\mathbf{x}^*)$ and $G(\mathbf{y}^*)$ will move to the two boundary grids, and this duration will last for at most $O\left(\frac{\mu N}{P_1(\varepsilon)}\right)$ expected generations. In what follows, we only consider the case that there are two vectors located in the two respective boundary grids of the Pareto front.
   (b) When $\min_{\substack{\mathbf{x}\neq\mathbf{y}\\ \mathbf{x},\mathbf{y}\in\mathcal{P}^{(t)}}} gdist(\mathbf{x},\mathbf{y}) < \lfloor\frac{k}{2}\rfloor - 1$, there are two points $\mathbf{x},\mathbf{y}$ with $gdist(\mathbf{x},\mathbf{y}) = \lfloor\frac{k}{2}\rfloor - 2$. Moreover, if there are no $\mathbf{x}_P \in PS\setminus\mathcal{P}$ with $gdist(\mathbf{x}_P,\mathcal{P}) = \lfloor\frac{k}{2}\rfloor$, it comes to the result that

$$\max_{\mathbf{x}\in\mathcal{P}} \min_{\substack{\mathbf{y}\neq\mathbf{x}\\ \mathbf{y}\in\mathcal{P}}} gdist(\mathbf{x},\mathbf{y}) < 2\left\lfloor\frac{k}{2}\right\rfloor,$$

which cannot hold because $(\mu-2)\times(2\times\lfloor\frac{k}{2}\rfloor) + \lfloor\frac{k}{2}\rfloor - 2$ is necessarily less than $N-1$. Thus, the population can evolve into a set of Pareto solutions with

$$\min_{\substack{\mathbf{x}\neq\mathbf{y}\\ \mathbf{x},\mathbf{y}\in\mathcal{P}^{(t)}}} gdist(\mathbf{x},\mathbf{y}) \geqslant \left\lfloor\frac{k}{2}\right\rfloor - 1,$$

and the result can be obtained as follows.

If there exists an $\mathbf{x}_P \in PS\setminus\mathcal{P}$ with $Gdist(\mathbf{x}_P,\mathcal{P}) = \lfloor\frac{k}{2}\rfloor$, then it is generated with probability $\frac{1}{\mu}P_{\lfloor\frac{k}{2}\rfloor}(\varepsilon)$, and $\mathbf{x}_P$ will replace the worst individual with $Dist(\mathbf{x}_w,\mathcal{P}\setminus\{\mathbf{x}\}) < \lfloor\frac{k}{2}\rfloor - 1$ in expected time $\mathcal{O}(\frac{\mu}{P_{\lfloor\frac{k}{2}\rfloor}(\varepsilon)})$. After at most μ updates, the distance between any two individuals is greater than or equal to $\lfloor\frac{k}{2}\rfloor - 1$, and the total expected runtime of this process is $\mathcal{O}(\frac{\mu^2}{P_{\lfloor\frac{k}{2}\rfloor}(\varepsilon)})$.
   (c) At the second step, the whole population evolves to an ε-adaptive Pareto front. Denoting $M = \min_{\mathbf{x},\mathbf{y}\in\mathcal{P}} dist(\mathbf{x},\mathbf{y})$, we have $M \leqslant k$. Then, if there exists a Pareto vector $\mathbf{x}_P \notin \mathcal{P}$ with $Gdist(x_P,\mathcal{P}) = M+2$, it is generated in expected time $\mathcal{O}(\frac{\mu}{P_{M+2}(\varepsilon)})$. After at most μ updates, the total expected time is $\mathcal{O}(\frac{\mu^2}{P_{M+2}(\varepsilon)})$, and the minimum grid distance is greater than or equal to $M+2$. Repeat this process if possible. Then, after at most a finite number of times, the minimum distance is k, and an ε-adaptive Pareto front is achieved in the expected runtime

$$\mathcal{O}\left(\frac{\mu^2}{P_{M+2}(\varepsilon)}\right) + \mathcal{O}\left(\frac{\mu^2}{P_{M+4}(\varepsilon)}\right) + \ldots + \mathcal{O}\left(\frac{\mu^2}{P_k(\varepsilon)}\right) = \mathcal{O}\left(\frac{\mu^2}{P_k(\varepsilon)}\right).$$

2. If $k < \frac{N-1}{\mu-1} < k+1, k = 4,5,\ldots$, a set of Pareto solutions $\mathcal{P}$ with $\min_{\mathbf{x},y\in\mathcal{P},\mathbf{x}\neq\mathbf{y} gdist(\mathbf{x},\mathbf{y})} = k$ is generated in the expected runtime $\mathcal{O}(\frac{\mu^2}{P_k(\varepsilon)})$. Then, in the expected runtime $\mathcal{O}(\frac{\mu^2}{P_1(\varepsilon)})$, an ε-adaptive Pareto front is obtained. That is to say, the expected runtime in this case is $\mathcal{O}(\frac{\mu^2}{P_k(\varepsilon)})$, too.

Thus, the $(\mu+1)$ MOEA obtains an ε-adaptive Pareto front of SCTOP in expected time $\mathcal{O}(\frac{\mu^2}{P_k(\varepsilon)})$. Moreover, because

$$\begin{aligned}
P_k(\varepsilon) = &\ \frac{1}{2\pi\sigma^2}\int_{(\lceil 1/\varepsilon\rceil-1)\varepsilon}^{1} dy_1 \int_{k\varepsilon/\sqrt{2}}^{(k+1)\varepsilon/\sqrt{2}} \exp\left\{-\frac{(y_1-x_1)^2+y_2^2}{2\sigma^2}\right\} dy_2 \\
\geqslant &\ \frac{1}{2\pi\sigma^2}\int_{0}^{1-(\lceil 1/\varepsilon\rceil-1)\varepsilon} dy_1 \int_{k\varepsilon/\sqrt{2}}^{(k+1)\varepsilon/\sqrt{2}} \exp\left\{-\frac{y_1^2+y_2^2}{2\sigma^2}\right\} dy_2 \\
= &\ \frac{1}{2\pi\sigma^2}\left((1-(\lceil 1/\varepsilon\rceil-1)\varepsilon) + o((1-(\lceil 1/\varepsilon\rceil-1)\varepsilon)^2)\right)\left(e^{-\frac{1}{2\sigma^2}(k\varepsilon/\sqrt{2})^2}\cdot\frac{\varepsilon}{\sqrt{2}} + o\left(\left(\frac{\varepsilon}{\sqrt{2}}\right)^2\right)\right) \\
= &\ \Omega\left(\frac{\varepsilon}{\sigma^2}e^{-\frac{2}{\sigma^2\mu^2}}\right),
\end{aligned}$$

the expected runtime of the $(\mu+1)$ MOEA to obtain an ε-Pareto front of SCTOP is $\mathcal{O}(\frac{\sigma^2\mu^2}{\varepsilon}e^{\frac{2}{\sigma^2\mu^2}})$. □

Table 1
Parameter settings for numerical experiments.

MOP	Number of bits	Population/archive size	Generations
LOTZ	20	21	2000
LF'_δ	36	19	1000
SCTOP	–	20	1000

Table 2
Statistical comparisons of IGD values for two algorithms.

Function	Algorithm	Best	Worst	Mean	St. dev.
LOTZ	NSGA-II	0.2694	0.6734	0.4674	0.0854
	$(\mu+\mu)$ MOEA	0	0.2694	0.1037	0.0612
LF'_δ	NSGA-II	1.0419	1.6787	1.3293	0.1457
	$(\mu+\mu)$ MOEA	0.8939	0.9517	0.9204	0.0109
SCTOP	NSGA-II	0.1410	0.3365	0.2413	0.0354
	$(\mu+\mu)$ MOEA	0.0151	0.2395	0.1121	0.0683

With respect to the approximation precision ε, Theorem 4 shows that the expected runtime is of the order of $\mathcal{O}(\frac{1}{\varepsilon})$. Then, when the approximation precision ε is set small, it is hard to locate the points whose distance to the adaptive Pareto front are less than ε, which leads to a long expected runtime of the $(\mu+1)$ MOEA. Regarding the mutation parameter σ and the population size μ, the expected runtime is $\mathcal{O}(\sigma^2\mu^2 e^{\frac{2}{\sigma^2\mu^2}})$. Generally speaking, the expected runtime is of the order of $\mathcal{O}(\sigma^2\mu^2)$, because $e^{\frac{2}{\sigma^2\mu^2}}$ is always less than one. However, $\sigma^2\mu^2 e^{\frac{2}{\sigma^2\mu^2}}$ will reach its minimum value when $\sigma\mu = \sqrt{2}$, which means the complexity order could be minimized when σ and μ are set to minimize $|\sigma\mu - \sqrt{2}|$.

4. Numerical results

In this section, we compare our results with those of the NSGA-II to demonstrate the efficiency of the proposed algorithm. Because the NSGA-II generates an intermediate population of size μ, we propose a variant of the $(\mu+1)$ MOEA, termed $(\mu+\mu)$ MOEA, which generates μ candidates at each generation, and updates the population by running Algorithm 4 μ times. By attaching an archive to the framework of NSGA-II, the generated candidates update the population of NSGA-II and the archive, respectively. In this way, at each generation the same candidates are generated for the two different updating strategies, which are the selection strategy of NSGA-II and the archive-updating strategy illustrated by Algorithm 4. For the discrete MOPs, the candidates are generated by bitwise mutation, and for the continuous SCTOP, the algorithm generates candidates by polynomial mutation [10].

Then, the comparison is performed by three MOPs for parameters listed in Table 1, and the Inverse General Distance (IGD) [38] is employed to assess performances. For the polynomial DMOP LOTZ, the reference set is the collection of all Pareto vectors, and for the Exponential DMOP LF'_δ and the SCTOP, adaptive Pareto fronts of the population size are taken as the reference sets.[3] After fifty independent runs, the statistical results of IGD values are collected in Table 2.

The numerical results show that the proposed $(\mu+\mu)$ MOEA is competitive with the NSGA-II for the three test MOPs, except that the $(\mu+\mu)$ MOEA standard deviation for SCTOP is slightly worse. The reason could be that the $(\mu+\mu)$ MOEA updates the individuals of population one by one and the NSGA-II generates the next population by selecting μ individuals from the union of the present population and the intermediate population from a global perspective. Consequently, the results of NSGA-II for simple CMOPs could be more stable, although the rest of the statistical results are all worse than those of the $(\mu+\mu)$ MOEA.

5. Conclusions and future work

Based on the definitions of an adaptive Pareto front and its ϵ-approximation, this paper presents a runtime analysis of an estimator-based $(\mu+1)$ MOEA for three MOPs. The theoretical results show that when the population size is less than the total number of Pareto vectors, the $(\mu+1)$ MOEA cannot achieve the expected polynomial runtime for the investigated DMOPs. Thus, we should set the population size equal to the total number of LOTZ. For the exponential DMOP LF'_δ, the expected polynomial runtime can be obtained by maintaining the ratio of $\frac{n}{2}$ to $\mu - 1$ over a proper interval. For the given CMOP with a continuous Pareto front, it is shown that the $(\mu+1)$ MOEA can efficiently solve it by obtaining an ε-adaptive Pareto

[3] Because the total number of Pareto vectors of LOTZ is $n+1$, we set the population/archive size to 21 to obtain all Pareto vectors of LOTZ. Moreover, an adaptive Pareto front of size $\frac{n}{2}+1$ can be uniformly distributed along the Pareto front, and accordingly, we set the population/archive size to 19 when the LF'_δ is considered.

front in the expected polynomial runtime. The numerical results also validate that the $(\mu + \mu)$ MOEA, a variant of the proposed $(\mu + 1)$ MOEA, can outperform NSGA-II for the three investigated MOPs.

Generally speaking, because the diversity estimator-based $(\mu + 1)$ MOEA can accommodate the Pareto front adaptively, it can solve the three MOPs under investigation efficiently. However, the fitness function defined based on the true Pareto front is unavailable when a MOP with unknown properties is encountered. Thus, from a practical point of view, an appropriate fitness function is critical for designing an efficient $(\mu + 1)$ MOEA. Moreover, to improve its global exploration ability, crossover must be introduced when the $(\mu + 1)$ MOEA is utilized in practical applications. Furthermore, the $(\mu + 1)$ MOEA can also be improved to generate λ individuals at each generation. Thus, future work will focus on studies of the $(\mu + \lambda)$ MOEAs, including time-complexity analysis of $(\mu + \lambda)$ MOEAs and the design of a high-performance MOEA inspired by the theoretical results.

Acknowledgments

This work was supported by the Chinese National Natural Science Foundation under Grants No. 61173060 and No. 61303028, the Major Research Plan of the National Natural Science Foundation of China under Grant No. 91230118 and the Fundamental Research Funds for the Central Universities (WUT:2013-Ia-001).

References

[1] R. Berghammer, T. Friedrich, F. Neumann, Convergence of set-based multi-objective optimization, indicators and deteriorative cycles, Theor. Comput. Sci. 45 (2012) 2–17.
[2] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume, Eur. J. Oper. Res. 181 (3) (2007) 1653–1669.
[3] N. Beume, M. Laumanns, G. Rudolph, Convergence rates of (1 + 1) evolutionary multiobjective optimization algorithms, in: Proc. of 11th Intenational Conference on Parallel Problem Solving From Nature, LNCS, vol. 6238, Springer, Berlin, 2010, pp. 597–606.
[4] N. Beume, M. Laumanns, G. Rudolph, Convergence rates of SMS-EMOA on continuous bi-objective problems classes, in: Proc. of Foundations of Genetic Algorithms, ACM, 2011, pp. 243–251.
[5] D. Brockhoff, T. Friedrich, F. Neumann, Analyzing hypervolume indicator based algorithms, in: Proc. of 10th Intenational Conference on Parallel Problem Solving From Nature, LNCS, vol. 5199, Springer, Berlin, 2008, pp. 651–660.
[6] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, E. Zitzler, On the effects of adding objectives to plateau functions, IEEE Trans. Circ. Syst. Vid. Technol. 13 (3) (2009) 591–603.
[7] C.A. Coello Coello, Evolutionary multi-objective optimization: some current research trends and topics that remain to be explored, Frontiers Comput. Sci. China 3 (2009) 18–30.
[8] Y. Chen, X. Zou, W. Xie, Convergence of multi-objective evolutionary algorithms to a uniformly distributed representation of the Pareto front, Inf. Sci. 181 (16) (2011) 3336–3355.
[9] C. Cheng, W. Wang, D. Xu, K.W. Chau, Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos, Water Resour. Manage. 22 (2008) 895–909.
[10] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[11] M. Emmerich, N. Beume, B. Naujoks, An EMO algorithm using the hypervolume measure as selection criterion, in: Proc. of 3rd Intenational Conference on Evolutionary Multi-Criterial Optimization, LNCS, vol. 3410, Springer, Berlin, 2005, pp. 62–76.
[12] T. Friedrich, C. Horoba, F. Neumann, Illustration of fairness in evolutionary multi-objetive optimization, Theor. Comput. Sci. 412 (17) (2011) 1546–1556.
[13] D.W. Gong, J. Sun, X.F. Ji, Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems, Inf. Sci. 233 (2013) 141–161.
[14] O. Giel, Expected runtimes of a simple multi-objective evolutionary algorithm, in: Proc. of 2003 IEEE Conference on Evolutionary Computation, IEEE, NJ, 2003, pp. 1918–1925.
[15] O. Giel, P.K. Lehre, On the effect of populations in evolutionary multi-objective optimisation, Evol. Comput. 18 (3) (2010) 335–356.
[16] T. Hanne, A multiobjective evolutionary algorithm for approximating the efficient set, Eur. J. Oper. Res. 176 (2007) 1723–1734.
[17] J. He, X. Yao, Drift analysis and average time complexity of evolutionary algorithms, Artif. Intel. 127 (1) (2001) 57–85.
[18] C. H oroba, F. N eumann, Benefits and drawbacks for the use of ϵ-dominance in evolutionary multi-objective optimization, in: Proc. of 2008 Genetic and Evolutionary Computation Conference, GECCO, ACM, 2008, pp. 641–648.
[19] C. Horoba, F. Neumann, Additive approximations of Pareto-optimal sets by evolutionary multi-objective algorithms, in: Proc. of Foundations of Genetic Algorithms, FOGA, ACM, 2009, pp. 79–86.
[20] C. Horoba, Exploring the runtime of an evolutionary algorithm for the multi-objective shortest path problem, Evol. Comput. 18 (3) (2010) 357–381.
[21] E.J. Hughes, Evolutionary many-objective optimisation: Many once or one many?, in: Proc of 2005 IEEE Conference on Evolutionary Computation, CEC, IEEE Press, 2005, pp. 222–227.
[22] W. Jia, B. Ling, K.W. Chau, L. Heutte, Palmprint identification using restricted fusion, Appl. Math. Comput. 205 (2008) 927–934.
[23] L.C. Jiao, H.D. Wang, R.H. Shang, F. Liu, A co-evolutionary multi-objective optimization algorithm based on direction vectors, Inf. Sci. 228 (2013) 90–112.
[24] J.D. Knowles, D.W. Corne, Approximating the nondominated front using the Pareto archived evolution strategy, Evol. Comput. 8 (2) (2000) 149–172.
[25] J.D. Knowles, D.W. Corne, Properties of an adaptive archiving algorithm for storing nondominated vectors, IEEE Trans. Evol. Comput. 7 (2) (2003) 100–116.
[26] M. Laumanns, L. Thiele, K. Deb, E. Zitzler, Combining convergence and diversity in evolutionary multiobjective optimization, Evol. Comput. 10 (3) (2002) 263–282.
[27] M. Laumanns, L. Thiele, E. Welzl, K. Deb, Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem, in: Proc. of 2002 International Conference on Parallel Problem Solving From Nature, LNCS, vol. 2439, Springer, 2002, pp. 44–53.
[28] M. Laumanns, L. Thiele, E. Zitzler, Running time analysis of multiobjective evolutionary algorithm on pseudo-boolean functions, IEEE Trans. Evol. Comput. 8 (2) (2004) 170–182.
[29] K. Li, S. Kwong, R. Wang, K.S. Tang, K.F. Man, Learning paradigm based on jumping genes: a general framework for enhancing exploration in evolutionary multiobjective optimization, Inf. Sci. 226 (2013) 1–22.
[30] J.Y. Lin, C.T. Cheng, K.W. Chau, Using support vector machines for long-term discharge prediction, Hydrol. Sci. J. 51 (2006) 599–612.
[31] F. Neumann, Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem, Eur. J. Oper. Res. 181 (2007) 1620–1629.

[32] O. Schütze, M. Laumanns, C.A. Coello Coello, Convergence of stochastic search algorithms to finite size Pareto set approximations, J. Global Optim. 41 (2008) 559–577.
[33] O. Schütze, A. Lara, C.A. Coello Coello, On the influence of the number of objectives on the hardness of a multiobjective optimization problem, IEEE Trans. Evol. Comput. 15 (4) (2011) 444–455.
[34] Y.Y. Tan, Y.C. Jiao, H. Li, X.K. Wang, A modification to MOEA/D-DE for multiobjective optimization problems with complicated Pareto sets, Inf. Sci. 213 (2012) 14–38.
[35] L.X. Tang, X.P. Wang, A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems, IEEE Trans. Evol. Comput. 17 (1) (2013) 20–45.
[36] C.L. Wu, K.W. Chau, Y.S. Li, Predicting monthly streamflow using data-driven models coupled with data-preprocessing techniques, Water Resour. Res. 45 (2009), http://dx.doi.org/10.1029/2007WR006737.
[37] J.X. Xie, C.T. Cheng, K.W. Chau, Y.Z. Pei, A hybrid adaptive time-delay neural network model for multi-step-ahead prediction of sunspot activity, Int. J. Environ. Pollut. 28 (2006) 364–381.
[38] Q. Zhang, H. Li, MOEA/D: a multi-objective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[39] J. Zhang, K.W. Chau, Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization, J. Universal Comput. Sci. 15 (2009) 840–858.
[40] Y. Zhou, J. He, Convergence analysis of a self-adaptive multi-objective evolutionary algorithm based on grids, Inf. Process. Lett. 104 (2007) 117–122.
[41] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271.
[42] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength Pareto evolutionary algorithm, TIK-Report No. 103. Zurich, Switzerland, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich.
[43] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, in: Proc. of 8th International Conference on Parallel Problem Solving from Nature, LNCS, vol. 886, Springer, Berlin, 2004, pp. 832–842.
[44] X. Zou, Y. Chen, M. Liu, L. Kang, A new evolutionary algorithm for solving many-objective optimization problems, IEEE Trans. Syst. Man Cyber. Part B 38 (5) (2008) 1402–1412.