Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18)

A General Approach to Running Time Analysis of Multi-objective Evolutionary Algorithms

Chao Bian[1], **Chao Qian**[1], **Ke Tang**[2]
[1] Anhui Province Key Lab of Big Data Analysis and Application,
University of Science and Technology of China, Hefei 230027, China
[2] Shenzhen Key Lab of Computational Intelligence, Department of Computer Science and Engineering,
Southern University of Science and Technology, Shenzhen 518055, China
biancht@mail.ustc.edu.cn, chaoqian@ustc.edu.cn, tangk3@sustc.edu.cn

Abstract

Evolutionary algorithms (EAs) have been widely applied to solve multi-objective optimization problems. In contrast to great practical successes, their theoretical foundations are much less developed, even for the essential theoretical aspect, i.e., running time analysis. In this paper, we propose a general approach to estimating upper bounds on the expected running time of multi-objective EAs (MOEAs), and then apply it to diverse situations, including bi-objective and many-objective optimization as well as exact and approximate analysis. For some known asymptotic bounds, our analysis not only provides their leading constants, but also improves them asymptotically. Moreover, our results provide some theoretical justification for the good empirical performance of MOEAs in solving multi-objective combinatorial problems.

1 Introduction

Multi-objective optimization, which requires optimizing several objective functions simultaneously, arises in many areas. Since the objective functions are usually conflicting, the goal is to find a set of Pareto optimal solutions (or the *Pareto front*), which represent different optimal trade-offs between objectives. Evolutionary algorithms (EAs) [Bäck, 1996] are a kind of randomized heuristic optimization algorithms, inspired by natural evolution. They maintain a set of solutions (called a population), and iteratively improve the population by genetic operators. Due to their population-based nature, EAs are popular for solving multi-objective optimization problems, and have been found well useful in many real-world applications [Coello Coello and Lamont, 2004].

However, due to their sophisticated behaviors of mimicking natural phenomena, the theoretical analysis of EAs is difficult. Much effort thus has been devoted to understanding the behavior of EAs from a theoretical point of view [Neumann and Witt, 2010; Auger and Doerr, 2011], but most of them focus on single-objective optimization. In fact, multi-objective EAs (MOEAs) are even more difficult to be analyzed owing to the hardness of multi-objective optimization.

To the best of our knowledge, only a few pieces of case-specific studies have been reported on the running time analysis of MOEAs. The running time complexity, which measures how fast an algorithm solves an optimization problem, is one essential theoretical aspect. In [Giel, 2003], the GSEMO (a simple MOEA with a global mutation operator) was proved to find the Pareto front of $\texttt{LOTZ}$ in $O(n^3)$ expected running time (where n is the problem size), while for another bi-objective problem $\texttt{COCZ}$, it needs $O(n^2 \log n)$ expected time [Qian *et al.*, 2013]. For $m\texttt{LOTZ}$ and $m\texttt{COCZ}$ with m objectives, which are generalized from $\texttt{LOTZ}$ and $\texttt{COCZ}$, both the expected running time of the SEMO (a counterpart of the GSEMO, but with a local mutation operator) were shown to be $O(n^{m+1})$ [Laumanns *et al.*, 2004]. More results on synthetic problems include [Friedrich *et al.*, 2010; Giel and Lehre, 2010; Friedrich *et al.*, 2011; Neumann, 2012; Doerr *et al.*, 2013; 2016; Qian *et al.*, 2016; Osuna *et al.*, 2017]. The analysis on NP-hard multi-objective combinatorial problems has been only slightly touched. For bi-objective minimum spanning trees, the GSEMO was proved able to find a 2-approximation of the Pareto front in expected pseudo-polynomial time [Neumann, 2007]; and for multi-objective shortest paths, a variant of the GSEMO can achieve an $(1+\epsilon)$-approximation in polynomial time [Horoba, 2009; Neumann and Theile, 2010].

Note that the analysis approaches employed in most of the above mentioned studies are case-specific, which cannot provide a general guidance to analyze the running time of a given MOEA solving a given problem. Meanwhile, ad hoc analyses starting from scratch are quite difficult. In this paper, we thus propose a general approach (**Theorem 1**) for estimating upper bounds on the running time of MOEAs. The idea is to contrast a given sophisticated MOEA process with an easily-analyzable process, instead of directly analyzing the MOEA process. We apply the approach to diverse situations, including bi-objective and many-objective optimization as well as exact and approximate analysis. The theoretical results are:

- For the GSEMO solving the two bi-objective problems $\texttt{LOTZ}$ and $\texttt{COCZ}$, the expected time for finding the Pareto front is at most $6n^3$ and $3n^2 \log n$ (**Theorems 2, 3**), respectively, which are consistent with the known asymptotic bounds $O(n^3)$ [Giel, 2003] and $O(n^2 \log n)$ [Qian *et al.*, 2013], and further give the leading constants.

- For the SEMO solving the scalable m-objective problem mCOCZ where $m \geq 4$, the expected time for finding the Pareto front is $O(n^m)$ (**Theorem 4**), which is tighter than the known bound $O(n^{m+1})$ [Laumanns *et al.*, 2004] by a factor of n. Furthermore, a better upper bound $O(n^3 \log n)$ is also derived for $m = 4$.
- For optimizing two linear functions simultaneously, which often appears in multi-objective combinatorial problems such as multi-objective minimum spanning trees and multi-objective knapsacks, a variant of the GSEMO can find a good approximation of the Pareto front in polynomial time (**Theorem 5**). Our analysis thus provides some theoretical justification for the good empirical performance of MOEAs in solving multi-objective combinatorial problems [Zhou and Gen, 1999; Ishibuchi *et al.*, 2015].

The rest of this paper starts with a section of preliminaries. Section 3 presents the proposed approach, which is applied in three subsequent sections to analyze MOEAs in different situations. Section 7 concludes the paper.

2 Preliminaries

2.1 Multi-objective Evolutionary Algorithms

Multi-objective optimization requires simultaneously optimizing two or more objective functions, as shown in Definition 1. We consider maximization here, while minimization can be defined similarly. The objectives are usually conflicting, and thus there is no canonical complete order in the solution space $\mathcal{S}$. The comparison between solutions relies on the *domination* relationship, as presented in Definition 2. A solution is *Pareto optimal* if there is no other solution in $\mathcal{S}$ that dominates it. The set of objective vectors of all the Pareto optimal solutions constitutes the *Pareto front*. The goal of multi-objective optimization is to find the Pareto front, that is, to find at least one corresponding solution for each objective vector in the Pareto front.

Definition 1 (Multi-objective Optimization)**.** *Given a feasible solution space $\mathcal{S}$ and objective functions $f_1, f_2, \ldots, f_m$, multi-objective optimization can be formulated as*

$$\max_{\boldsymbol{s} \in \mathcal{S}} \ \big(f_1(\boldsymbol{s}), f_2(\boldsymbol{s}), ..., f_m(\boldsymbol{s})\big).$$

Definition 2 (Domination)**.** *Let $\boldsymbol{f} = (f_1, f_2, \ldots, f_m) : \mathcal{S} \to \mathbb{R}^m$ be the objective vector. For two solutions $\boldsymbol{s}$ and $\boldsymbol{s}' \in \mathcal{S}$:*

1. $\boldsymbol{s}$ weakly dominates $\boldsymbol{s}'$ *if* $\forall 1 \leq i \leq m, f_i(\boldsymbol{s}) \geq f_i(\boldsymbol{s}')$, *denoted as* $\boldsymbol{s} \succeq \boldsymbol{s}'$;
2. $\boldsymbol{s}$ dominates $\boldsymbol{s}'$ *if* $\boldsymbol{s} \succeq \boldsymbol{s}'$ *and* $f_i(\boldsymbol{s}) > f_i(\boldsymbol{s}')$ *for some* i, *denoted as* $\boldsymbol{s} \succ \boldsymbol{s}'$.

The GSEMO algorithm is a simple MOEA for multi-objective optimization over the Boolean solution space $\mathcal{S} = \{0,1\}^n$. As described in Algorithm 1, it randomly selects an initial solution, and then repeatedly tries to improve the population P. In each iteration, a solution uniformly selected from the current P is used to generate a new solution by bit-wise mutation; then the newly generated solution is compared with the solutions in P, and only non-dominated solutions are kept. The SEMO algorithm is the same as the GSEMO except that bit-wise mutation which searches globally is replaced by one-bit mutation which searches locally, i.e., line 5 of Algorithm 1 becomes "Create $\boldsymbol{s}'$ by flipping a randomly chosen bit of $\boldsymbol{s}$". These two algorithms explain the common structure of various MOEAs and are widely used in theoretical analyses [Laumanns *et al.*, 2004; Friedrich *et al.*, 2010; Qian *et al.*, 2013]. We will also use them in case studies.

Algorithm 1 GSEMO Algorithm

Given the solution space $\mathcal{S} = \{0,1\}^n$ and the objective vector $\boldsymbol{f}$, the procedure:

1: Choose $\boldsymbol{s} \in \mathcal{S}$ uniformly at random.
2: $P \leftarrow \{\boldsymbol{s}\}$.
3: **repeat until** some criterion is met
4: Choose $\boldsymbol{s}$ from P uniformly at random.
5: Create $\boldsymbol{s}'$ by flipping each bit of $\boldsymbol{s}$ with prob. $1/n$.
6: **if** $\nexists \boldsymbol{z} \in P$ such that $\boldsymbol{z} \succ \boldsymbol{s}'$
7: $P \leftarrow (P \setminus \{\boldsymbol{z} \in P \mid \boldsymbol{s}' \succeq \boldsymbol{z}\}) \cup \{\boldsymbol{s}'\}$.

The running time of a MOEA is usually measured by the number of calls to $\boldsymbol{f}$ (the most costly computational process) until finding the Pareto front or an approximation of the Pareto front. For any $c \leq 1$, a set P of solutions is a c-approximation of the Pareto front if for each objective vector $(f_1^*, f_2^*, \ldots, f_m^*)$ in the Pareto front, there always exists a solution $\boldsymbol{s} \in P$ such that $\forall i \in \{1, 2, \ldots, m\} : f_i(\boldsymbol{s}) \geq c \cdot f_i^*$.

2.2 Markov Chain Modeling

EAs often generate offspring solutions from their current solutions rather than the historical ones; thus, they can be modeled as Markov chains. Let $\mathcal{X}$ be the population space and $\mathcal{X}^* \subseteq \mathcal{X}$ be the target population space. If the goal is to find a c-approximation of the Pareto front, each population in $\mathcal{X}^*$ is a c-approximation of the Pareto front. Let $\xi_t \in \mathcal{X}$ be the population after t generations. Then, a MOEA can be described as a random sequence $\{\xi_0, \xi_1, \xi_2, \ldots\}$. Since ξ_{i+1} can often be decided from ξ_i (i.e., $\mathrm{P}(\xi_{i+1}|\xi_i, \ldots, \xi_0) = \mathrm{P}(\xi_{i+1}|\xi_i)$), the random sequence forms a Markov chain $\{\xi_t\}_{t=0}^{+\infty}$ with state space $\mathcal{X}$, denoted as "$\xi \in \mathcal{X}$" for simplicity.

Given a Markov chain $\xi \in \mathcal{X}$ and $\xi_{t_0} = x$, we define τ as a random variable such that $\tau = \min\{t \geq 0 \mid \xi_{t_0+t} \in \mathcal{X}^*\}$. That is, τ is the number of steps needed to reach the target space for the first time. The mathematical expectation of τ, $\mathbb{E}[\![\tau|\xi_{t_0} = x]\!] = \sum_{i=0}^{\infty} i \cdot \mathrm{P}(\tau = i|\xi_{t_0} = x)$, is called the *conditional first hitting time* (CFHT). If ξ_{t_0} is drawn from a distribution π_{t_0}, the expectation of the CFHT over π_{t_0}, $\mathbb{E}[\![\tau|\xi_{t_0} \sim \pi_{t_0}]\!] = \sum_{x \in \mathcal{X}} \pi_{t_0}(x)\mathbb{E}[\![\tau|\xi_{t_0} = x]\!]$, is called the *distribution-CFHT* (DCFHT). For a Markov chain $\xi \in \mathcal{X}$ modeling a MOEA, $\mathbb{E}[\![\tau|\xi_0 \sim \pi_0]\!]$ is just the expected number of iterations of the MOEA until reaching $\mathcal{X}^*$. We always use $\mathbb{E}[\![\cdot]\!]$ to denote the expectation of a random variable.

A Markov chain $\xi \in \mathcal{X}$ is absorbing, if $\forall t \geq 0 : \mathrm{P}(\xi_{t+1} \in \mathcal{X}^*|\xi_t \in \mathcal{X}^*) = 1$. Note that a Markov chain can be transformed to be absorbing by making it unchanged once finding a target state, which obviously does not affect its first hitting time. Lemma 1 is to compare the DCFHT of two absorbing Markov chains, which will be used in this paper.

Lemma 1. *[Yu et al., 2015] Given two absorbing Markov chains $\xi \in \mathcal{X}$ and $\xi' \in \mathcal{Y}$ with target spaces $\mathcal{X}^*$ and $\mathcal{Y}^*$, respectively, let τ and τ' denote their hitting times, respectively, and let π_t denote the distribution of ξ_t. Given a series of values $\{\rho_t \in \mathbb{R}\}_{t=0}^{+\infty}$ with $\rho = \sum_{t=0}^{+\infty} \rho_t$ and a mapping $\phi : \mathcal{X} \to \mathcal{Y}$ with $\forall x \in \mathcal{X} \setminus \mathcal{X}^* : \phi(x) \notin \mathcal{Y}^*$, if for all $t \geq 0$,*

$$\sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_t(x)\mathrm{P}(\xi_{t+1} \in \phi^{-1}(y)|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = y]\!]$$
$$\leq \sum_{u,y\in\mathcal{Y}} \pi_t^{\phi}(u)\mathrm{P}(\xi'_1 = y|\xi'_0 = u)\mathbb{E}[\![\tau'|\xi'_1 = y]\!] + \rho_t,$$

where $\phi^{-1}(y) = \{x \in \mathcal{X} \mid \phi(x) = y\}$, $\pi_t^{\phi}(u) = \sum_{x\in\phi^{-1}(u)} \pi_t(x)$, then $\mathbb{E}[\![\tau|\xi_0 \sim \pi_0]\!] \leq \mathbb{E}[\![\tau'|\xi'_0 \sim \pi_0^{\phi}]\!] + \rho$.

3 The Proposed Approach

For a MOEA solving a multi-objective problem, we propose a general approach (i.e., Theorem 1) to analyze running time upper bounds for finding a c-approximation of the Pareto front. Note that $c = 1$ implies finding the Pareto front. The idea is to model the given MOEA process as a Markov chain $\xi \in \mathcal{X}$, and compare it with an easy-to-analyze chain $\xi' \in \mathcal{Y}$. From the condition Eq. (1) of Theorem 1, we can see that the long-term behavior of the given chain $\xi \in \mathcal{X}$ is waived, since $\mathbb{E}[\![\tau|\xi_t]\!]$ is not involved. In the comparison, we need a function $h_{\boldsymbol{\alpha},c}$ to measure the goodness of $x \in \mathcal{X}$. As shown in Definition 3, $h_{\boldsymbol{\alpha},c}(x)$ is the sum of two terms: the maximal weighted sum of objective values of the solution in x, and the number of objective vectors in F^* which are c-approximated by x. It is easy to verify that $g(F^*, x, c) \leq |F^*|$, and for any x which is a c-approximation of F^*, $g(F^*, x, c) = |F^*|$. When $\boldsymbol{\alpha}$ and c are clear, we will write h for short.

Definition 3. *Given a Markov chain $\xi \in \mathcal{X}$ modeling a MOEA solving a multi-objective problem $(f_1, f_2, \ldots, f_m)$ with the Pareto front F^*, let the target space $\mathcal{X}^* = \{x \in \mathcal{X} \mid x$ is a c-approximation of $F^*\}$, where $c \leq 1$. For a non-negative real vector $\boldsymbol{\alpha} = (\alpha_0, \alpha_1, ..., \alpha_m)$, we define a function $h_{\boldsymbol{\alpha},c} : \mathcal{X} \to \mathbb{R}$ as for any $x \in \mathcal{X}$,*

$$h_{\boldsymbol{\alpha},c}(x) = \max_{\boldsymbol{s}\in x} \sum_{i=0}^{m} \alpha_i f_i(\boldsymbol{s}) + g(F^*, x, c),$$

where $f_0(\boldsymbol{s}) = 1$, $g(F^*, x, c) = |\{(f_1^*, f_2^*, \ldots, f_m^*) \in F^* \mid \exists \boldsymbol{s} \in x, \forall 1 \leq i \leq m : f_i(\boldsymbol{s}) \geq c \cdot f_i^*\}|$ and $|\cdot|$ denotes the cardinality of a set. If $\forall x \in \mathcal{X}$, $h_{\boldsymbol{\alpha},c}(x) \in \mathbb{N}_0$ and

$$\max_{x\in\mathcal{X}\setminus\mathcal{X}^*} h_{\boldsymbol{\alpha},c}(x) < \min_{x\in\mathcal{X}^*} h_{\boldsymbol{\alpha},c}(x),$$

then $h_{\boldsymbol{\alpha},c}$ is called a well-defined function.

Theorem 1. *Given a Markov chain $\xi \in \mathcal{X}$ modeling a MOEA solving a multi-objective problem with the Pareto front F^*, let the target space $\mathcal{X}^* = \{x \in \mathcal{X} | x$ is a c-approximation of $F^*\}$, where $c \leq 1$. If there exists a well-defined function $h_{\boldsymbol{\alpha},c} : \mathcal{X} \to \mathbb{N}_0$ and a Markov chain $\xi' \in \mathcal{Y} = \{0,1\}^r$ with $\mathcal{Y}^* = \{1^r\}$ satisfying that $\forall x \notin \mathcal{X}^*$, $\forall t \geq 0$,*

$$\sum_{i\in[r]} \mathrm{P}(\min\{h(\xi_{t+1}), r\} = i|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = 1^i0^{r-i}]\!] \quad (1)$$
$$\leq \sum_{y\in\mathcal{Y}} \mathrm{P}(\xi'_1 = y|\xi'_0 = 1^{h(x)}0^{r-h(x)})\mathbb{E}[\![\tau'|\xi'_1 = y]\!] + \delta,$$

where $r = \min\{h(x)|x \in \mathcal{X}^*\}$, $[r]$ denotes $\{0, \ldots, r\}$ and $\delta < 1$, the expected number of iterations until the MOEA finding a c-approximation of F^* when starting from x_0 is at most

$$\mathbb{E}[\![\tau'|\xi'_0 = 1^{\min\{h(x_0),r\}}0^{r-\min\{h(x_0),r\}}]\!]/(1 - \delta).$$

Proof. We use Lemma 1 to prove it by comparing the two Markov chains $\xi \in \mathcal{X}$ and $\xi' \in \mathcal{Y}$. We construct the mapping function $\phi : \mathcal{X} \to \mathcal{Y}$ as $\forall x \in \mathcal{X}$, $\phi(x) = 1^{\min\{h(x),r\}}0^{r-\min\{h(x),r\}}$. Since h is well-defined, we have $x \in \mathcal{X}^*$ iff $\phi(x) \in \mathcal{Y}^* = \{1^r\}$. Let π_t denote the distribution of ξ_t. Then, for any t,

$$\sum_{x\in\mathcal{X},y\in\mathcal{Y}} \pi_t(x)\mathrm{P}(\xi_{t+1} \in \phi^{-1}(y)|\xi_t = x))\mathbb{E}[\![\tau'|\xi'_0 = y]\!]$$
$$= \sum_{x\notin\mathcal{X}^*,i\in[r]} \pi_t(x)\mathrm{P}(\min\{h(\xi_{t+1}), r\} = i|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = 1^i0^{r-i}]\!]$$
$$\leq \sum_{x\notin\mathcal{X}^*} \pi_t(x)\Big(\sum_{y\in\mathcal{Y}} \mathrm{P}(\xi'_1 = y|\xi'_0 = 1^{h(x)}0^{r-h(x)})\mathbb{E}[\![\tau'|\xi'_1 = y]\!] + \delta\Big)$$
$$= \sum_{u,y\in\mathcal{Y}} \pi_t^{\phi}(u)\mathrm{P}(\xi'_1 = y|\xi'_0 = u)\mathbb{E}[\![\tau'|\xi'_1 = y]\!] + \delta \cdot (1 - \pi_t(\mathcal{X}^*)),$$

where the first equality is by the absorbing property of $\xi \in \mathcal{X}$ and the definition of ϕ, the inequality is by Eq. (1), and the last equality is by the absorbing property of $\xi' \in \mathcal{Y}$ and the definition of π_t^{ϕ} in Lemma 1. Thus, the condition of Lemma 1 holds with $\rho_t = \delta \cdot (1 - \pi_t(\mathcal{X}^*))$. When the MOEA starts from x_0, i.e., $\pi_0(x_0) = 1$, we get from Lemma 1 that

$$\mathbb{E}[\![\tau|\xi_0 = x_0]\!] \leq \mathbb{E}[\![\tau'|\xi'_0 = \phi(x_0)]\!] + \delta \cdot \sum_{t=0}^{+\infty}(1 - \pi_t(\mathcal{X}^*)).$$

By applying $\sum_{t=0}^{+\infty}(1 - \pi_t(\mathcal{X}^*)) = \mathbb{E}[\![\tau|\xi_0 = x_0]\!]$ and $\phi(x_0) = 1^{\min\{h(x_0),r\}}0^{r-\min\{h(x_0),r\}}$ to the above inequality, the theorem holds. □

Using the above approach to derive running time upper bounds of a MOEA (modeled by a chain $\xi \in \mathcal{X}$) for finding a c-approximation of the Pareto front, one needs to

1. design a well-defined function $h_{\boldsymbol{\alpha},c} : \mathcal{X} \to \mathbb{N}_0$ and construct a reference chain $\xi' \in \mathcal{Y} = \{0,1\}^r$ with $\mathcal{Y}^* = \{1^r\}$;
2. analyze the transition probabilities (i.e., $\mathrm{P}(\xi_{t+1}|\xi_t)$ and $\mathrm{P}(\xi'_{t+1}|\xi'_t)$) and the CFHT of the chain $\xi' \in \mathcal{Y}$ (i.e., $\mathbb{E}[\![\tau'|\xi'_t]\!]$), and then examine Eq. (1) to get the value of δ;
3. divide $\mathbb{E}[\![\tau'|\xi'_0]\!]$ by $1-\delta$ to get an upper bound on $\mathbb{E}[\![\tau|\xi_0]\!]$.

Since the running time of a MOEA is measured by the number of fitness evaluations, the cost of each iteration should be considered. For example, the expected running time of the GSEMO (i.e., Algorithm 1) is just the expected number of iterations (i.e, $\mathbb{E}[\![\tau|\xi_0]\!]$), since we only need to evaluate the newly generated solution $\boldsymbol{s}'$ in each iteration.

4 Bi-objective Analysis

In this section, we apply the proposed approach to analyze the running time of the GSEMO solving two bi-objective pseudo-Boolean problems `LOTZ` and `COCZ`, which are widely used in MOEAs' theoretical analyses. The derived running time bounds in Theorems 2 and 3 are consistent with the known asymptotic bounds $O(n^3)$ [Giel, 2003] and $O(n^2 \log n)$ [Qian *et al.*, 2013], and further give the leading constants. For `LOTZ` as presented in Definition 4, it is to maximize the number of leading 1-bits and trailing 0-bits of a solution $\boldsymbol{s} \in \{0,1\}^n$ simultaneously. The Pareto front is $F^* = \{(i, n-i) \mid 0 \leq i \leq n\}$, and the set of all the Pareto optimal solutions is $\mathcal{S}^* = \{0^n, 10^{n-1}, \ldots, 1^{n-1}0, 1^n\}$.

Definition 4 (LOTZ [Giel, 2003]). *The pseudo-Boolean function LOTZ:* $\{0,1\}^n \to \mathbb{N}_0^2$ *is defined as*

$$\mathrm{LOTZ}(\boldsymbol{s}) = \left(\sum_{i=1}^{n}\prod_{j=1}^{i} s_j, \sum_{i=1}^{n}\prod_{j=i}^{n}(1-s_j)\right),$$

where $s_j \in \{0,1\}$ *is the j-th bit of* $\boldsymbol{s}$.

Theorem 2. *For the GSEMO solving LOTZ, the expected running time for finding the Pareto front is at most* $6n^3$.

Proof. We use Theorem 1 to prove it. Let $\xi \in \mathcal{X}$ model the GSEMO solving LOTZ. From the algorithm procedure, we know that the solutions in any $x \in \mathcal{X}$ (i.e., in any possible population P) are incomparable. Let $c=1$ (i.e., the goal is to find the Pareto front F^*). Then $\mathcal{X}^* = \{\mathcal{S}^*\}$. We design the function h by setting $\boldsymbol{\alpha}=(0,1,1)$, i.e.,

$$h(x) = \max_{\boldsymbol{s}\in x}(f_1(\boldsymbol{s}) + f_2(\boldsymbol{s})) + |x \cap \mathcal{S}^*|.$$

Note that $g(F^*, x, 1)$ in Definition 3 equals to $|x \cap \mathcal{S}^*|$ here. For any $x \in \mathcal{X} \setminus \mathcal{X}^*$, $\max_{\boldsymbol{s}\in x}(f_1(\boldsymbol{s})+f_2(\boldsymbol{s})) \le n$ and $|x \cap \mathcal{S}^*| < n+1$, thus $h(x) < 2n+1 = h(\mathcal{S}^*)$. It is easy to see that $h(x) \in \mathbb{N}_0$. Thus, h is well-defined and $r = 2n+1$. For the Markov chain $\xi' \in \mathcal{Y} = \{0,1\}^r$ with $\mathcal{Y}^* = \{1^r\}$, we construct its transition behavior as follows: for any $\xi'_t \notin \mathcal{Y}^*$, ξ'_{t+1} is generated by flipping the first 0-bit of ξ'_t with probability $1/r$; otherwise, $\xi'_{t+1} = \xi'_t$. Thus, for any $\xi'_t \in \mathcal{Y}$, $\mathbb{E}[\![\tau'|\xi'_t]\!] = r \cdot |\xi'_t|_0$, where $|\cdot|_0$ denotes the number of 0-bits.

Then, we are to investigate Eq. (1). For any $\xi_t = x \notin \mathcal{X}^*$, assume that $h(x) = k < r$ and let $\hat{\boldsymbol{s}} \in \arg\max_{\boldsymbol{s}\in x}(f_1(\boldsymbol{s}) + f_2(\boldsymbol{s}))$. It is easy to verify that $h(\xi_{t+1}) \ge h(\xi_t) = k$, since $|x \cap \mathcal{S}^*|$ never decreases and if $\hat{\boldsymbol{s}}$ is deleted, the newly included solution $\boldsymbol{s'}$ must weakly dominate $\hat{\boldsymbol{s}}$ (line 7 of Algorithm 1) and thus $f_1(\boldsymbol{s'}) + f_2(\boldsymbol{s'}) \ge f_1(\hat{\boldsymbol{s}}) + f_2(\hat{\boldsymbol{s}})$. We then show that $h(\xi_{t+1}) \ge h(\xi_t) + 1 = k+1$ with probability at least $\frac{1}{en^2}$ by considering two cases. (1) If $|x \cap \mathcal{S}^*| = 0$, $\hat{\boldsymbol{s}}$ must not be Pareto optimal, and flipping only its first 0-bit will generate a new solution $\boldsymbol{s'}$ with $f_1(\boldsymbol{s'})+f_2(\boldsymbol{s'}) > f_1(\hat{\boldsymbol{s}})+f_2(\hat{\boldsymbol{s}}) = k$. (2) If $|x \cap \mathcal{S}^*| > 0$, flipping only the last 1-bit or the first 0-bit of a specific Pareto optimal solution in $x \cap \mathcal{S}^*$ can generate a new Pareto optimal solution, and thus $|\xi_{t+1} \cap \mathcal{S}^*| = |x \cap \mathcal{S}^*| + 1$. In both cases, $h(\xi_{t+1}) \ge h(x) + 1 = k+1$. Note that the probability of selecting a specific solution from x in line 4 of Algorithm 1 is $\frac{1}{|x|}$ and the probability of flipping only one specific bit in line 5 is $\frac{1}{n}(1-\frac{1}{n})^{n-1} \ge \frac{1}{en}$. Since the solutions in x are incomparable, we have $|x| \le n$. Thus, our claim that $\mathrm{P}(h(\xi_{t+1}) \ge k+1|\xi_t = x) \ge \frac{1}{en^2}$ holds. We then have

$$\begin{aligned}
&\sum_{i\in[r]} \mathrm{P}(\min\{h(\xi_{t+1}), r\} = i|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = 1^i 0^{r-i}]\!] \quad (2)\\
&\le \mathrm{P}(h(\xi_{t+1}) \ge k+1|\xi_t = x)\mathbb{E}[\![\tau'|\xi'_0 = 1^{k+1}0^{r-k-1}]\!]\\
&\quad + (1 - \mathrm{P}(h(\xi_{t+1}) \ge k+1|\xi_t = x))\mathbb{E}[\![\tau'|\xi'_0 = 1^k 0^{r-k}]\!]\\
&\le (1/(en^2)) \cdot r(r-k-1) + (1 - 1/(en^2)) \cdot r(r-k),
\end{aligned}$$

where the first inequality is by $\min\{h(\xi_{t+1}), r\} = h(\xi_{t+1})$, $h(\xi_{t+1}) \ge k$ and $\mathbb{E}[\![\tau'|\xi'_0 = 1^i 0^{r-i}]\!] = r(r-i)$ decreasing with i, and the second is by $\mathrm{P}(h(\xi_{t+1}) \ge k+1|\xi_t = x) \ge \frac{1}{en^2}$. Meanwhile, by the transition behavior of $\xi' \in \mathcal{Y}$, we get

$$\begin{aligned}
&\textstyle\sum_{y\in\mathcal{Y}} \mathrm{P}(\xi'_1 = y|\xi'_0 = 1^k 0^{r-k})\mathbb{E}[\![\tau'|\xi'_1 = y]\!]\\
&= (1/r) \cdot r(r-k-1) + (1-1/r) \cdot r(r-k).
\end{aligned} \tag{3}$$

By combining the above two formulas, the condition Eq. (1) of Theorem 1 holds with $\delta = 1 - \frac{r}{en^2} < 1 - \frac{2}{en}$.

Thus, we get that starting from any $x_0 \in \mathcal{X}$, the expected number of iterations until finding the Pareto front is at most $\mathbb{E}[\![\tau'|\xi'_0 = 1^{h(x_0)}0^{r-h(x_0)}]\!]/(1-\delta) \le \mathbb{E}[\![\tau'|\xi'_0 = 0^r]\!] \cdot \frac{en}{2} = (2n+1)^2 \cdot \frac{en}{2} \le 6n^3$. Since only a new solution is evaluated in each iteration of the GSEMO, the theorem holds. □

For COCZ as presented in Definition 5, the first objective is to maximize the number of 1-bits, and the other is to maximize the number of 1-bits in the first half of a solution plus the number of 0-bits in the second half. The Pareto front is $\{(n/2+i, n-i) \mid 0 \le i \le n/2\}$, and the set of all the Pareto optimal solutions is $\mathcal{S}^* = \{\boldsymbol{s} \in \{0,1\}^n \mid \sum_{i=1}^{n/2} s_i = n/2\}$.

Definition 5 (COCZ [Qian *et al.*, 2013]). *The pseudo-Boolean function COCZ:* $\{0,1\}^n \to \mathbb{N}_0^2$ *is defined as*

$$\mathrm{COCZ}(\boldsymbol{s}) = \left(\sum_{i=1}^{n} s_i, \sum_{i=1}^{n/2} s_i + \sum_{i=n/2+1}^{n}(1-s_i)\right),$$

where n is even.

Theorem 3. *For the GSEMO solving COCZ, the expected running time for finding the Pareto front is at most* $3n^2 \log n$.

Proof. Let $\xi \in \mathcal{X}$ model the GSEMO solving COCZ and $c=1$. Then, $\mathcal{X}^* = \{\{\boldsymbol{s^0}, \dots, \boldsymbol{s^{n/2}}\} | \boldsymbol{s^i} \in \mathcal{S}^*, \sum_{j=\frac{n}{2}+1}^{n} s_j^i = i\}$. We design the function h by setting $\boldsymbol{\alpha}=(-n/4, 1/2, 1/2)$, i.e.,

$$h(x) = \max_{\boldsymbol{s}\in x}(f_1(\boldsymbol{s}) + f_2(\boldsymbol{s}) - n/2)/2 + |x \cap \mathcal{S}^*|.$$

It is easy to verify that h is well-defined and $r=n+1$. For the chain $\xi' \in \mathcal{Y} = \{0,1\}^r$ with $\mathcal{Y}^* = \{1^r\}$, the transition behavior is constructed as follows. Suppose $\xi'_t = \boldsymbol{s}$ and $\sum_{i=1}^{n/2} s_i = j$. ξ'_{t+1} is generated by flipping the first 0-bit of $\boldsymbol{s}$ with probability $\frac{n/2-j}{r}$ for $j < \frac{n}{2}$ and $\frac{|\boldsymbol{s}|_0}{r}$ for $j = \frac{n}{2}$; otherwise, $\xi'_{t+1} = \xi'_t$. Thus, $\mathbb{E}[\![\tau'|\xi'_0 = 1^k 0^{r-k}]\!] - \mathbb{E}[\![\tau'|\xi'_0 = 1^{k+1}0^{r-k-1}]\!] = \frac{r}{n/2-k}$ for $k < \frac{n}{2}$, and $\frac{r}{n+1-k}$ for $\frac{n}{2} \le k < r$.

We then show that Eq. (1) holds with $\delta \le 1 - \frac{2}{e(n+1)}$. For any $\xi_t = x \notin \mathcal{X}^*$, assume that $h(x) = k < r$ and let $\hat{\boldsymbol{s}} \in \arg\max_{\boldsymbol{s}\in x}(f_1(\boldsymbol{s})+f_2(\boldsymbol{s}))$. It is easy to verify that $h(\xi_{t+1}) \ge h(\xi_t) = k$. We consider two cases. (1) If $|x \cap \mathcal{S}^*| = 0$, $\hat{\boldsymbol{s}}$ must not be Pareto optimal and $k < n/2$. Flipping only one 0-bit in the first half of $\hat{\boldsymbol{s}}$ will generate a new solution $\boldsymbol{s'}$ with $f_1(\boldsymbol{s'})+f_2(\boldsymbol{s'}) = f_1(\hat{\boldsymbol{s}})+f_2(\hat{\boldsymbol{s}})+2$. Thus, $\mathrm{P}(h(\xi_{t+1}) \ge k+1) \ge \frac{n/2-k}{en|x|}$. By using the same analysis as Eqs. (2) and (3), we get

$$\delta = \left(\frac{n/2-k}{en|x|} - \frac{n/2-k}{r}\right) \cdot \left(-\frac{r}{n/2-k}\right) \le 1 - \frac{2}{e(n+1)},$$

where the inequality is by $r = n+1$ and $|x| \le n/2+1$. (2) If $|x \cap \mathcal{S}^*| > 0$, we have $k = n/2 + |x \cap \mathcal{S}^*| \ge n/2+1$. Let $\hat{C} \in \arg\max_{C \subseteq x\cap\mathcal{S}^*} |C|$ s.t. $\max_{\boldsymbol{s}\in C} f_1(\boldsymbol{s}) - \min_{\boldsymbol{s}\in C} f_1(\boldsymbol{s}) = |C| - 1$. Let $\boldsymbol{u} = \arg\max_{\boldsymbol{s}\in\hat{C}} f_1(\boldsymbol{s})$ and $\boldsymbol{v} = \arg\min_{\boldsymbol{s}\in\hat{C}} f_1(\boldsymbol{s})$. Then, flipping one 0-bit of $\boldsymbol{u}$ or flipping one 1-bit in the second half of $\boldsymbol{v}$ can generate a new Pareto optimal solution. We thus have $\mathrm{P}(h(\xi_{t+1}) = k+1) \ge \frac{n-|\boldsymbol{u}|_1+|\boldsymbol{v}|_1-n/2}{en|x|} \ge \frac{n/2+1-|x\cap\mathcal{S}^*|}{en|x|} = \frac{n+1-k}{en|x|}$. Using the same analysis as case (1), we also get $\delta \le 1 - \frac{2}{e(n+1)}$.

By applying $\delta \le 1 - \frac{2}{e(n+1)}$ and $\mathbb{E}[\![\tau'|\xi'_0 = 0^r]\!] \le 2r(\log(n/2+1)+1)$ to Theorem 1, the theorem holds. □

5 Many-objective Analysis

In this section, we apply the proposed approach to analyze the running time of the SEMO solving the scalable m-objective problem mCOCZ, where $m \geq 4$. Note that as the number of objectives increases, the size of the Pareto front can exponentially increase and thus the problem becomes more complicated. Our derived running time bound in Theorem 4 is asymptotically tighter than the known bound $O(n^{m+1})$ [Laumanns *et al.*, 2004]. As presented in Definition 6, all the m objectives are cooperative in the first half of a solution (i.e., maximizing the number of 1-bits), and the second half is divided into $m' = m/2$ blocks, each of which is to maximize the number of 1-bits and 0-bits simultaneously. The Pareto front is $F^* = \{(\frac{n}{2}+i_1, \frac{n}{2}+n'-i_1, \dots, \frac{n}{2}+i_{m'}, \frac{n}{2}+n'-i_{m'}) \mid \forall 1 \leq j \leq m' : 0 \leq i_j \leq n'\}$, where $n' = n/m$ is the size of each block. The set of all the Pareto optimal solutions is $\mathcal{S}^* = \{\boldsymbol{s} \in \{0,1\}^n \mid \sum_{i=1}^{n/2} s_i = n/2\}$. It is easy to see that $|F^*| = (n'+1)^{m'}$ and $|\mathcal{S}^*| = 2^{n/2}$.

Definition 6 (mCOCZ [Laumanns *et al.*, 2004]). *The pseudo-Boolean function mCOCZ: $\{0,1\}^n \to \mathbb{N}_0^m$ is defined as*

$$m\text{COCZ}(\boldsymbol{s}) = (f_1(\boldsymbol{s}), f_2(\boldsymbol{s}), ..., f_m(\boldsymbol{s})),$$

where

$$f_k(\boldsymbol{s}) = \sum_{i=1}^{n/2} s_i + \begin{cases} \sum_{i=1}^{n'} s_{n/2+(k-1)n'/2+i}, & \text{if } k \text{ is odd,} \\ \sum_{i=1}^{n'} (1 - s_{n/2+(k-2)n'/2+i}), & \text{else,} \end{cases}$$

$m = 2 \cdot m'$, $n = m \cdot n'$ *and* $m', n' \in \mathbb{N}_0$.

Theorem 4. *For the SEMO solving mCOCZ, the expected running time for finding the Pareto front is $O(n^m)$ for $m > 4$ and $O(n^3 \log n)$ for $m = 4$.*

Proof. According to the intermediate result in Theorem 11 of [Laumanns *et al.*, 2004], i.e., the number of mutations allocated to non-Pareto-optimal solutions is $O(n^{m/2+1} \log n)$, we only need to consider the number of mutations allocated to Pareto optimal solutions, and thus we can assume that after $O(n^{m/2+1} \log n)$ iterations in expectation, the population will always contain only Pareto optimal solutions. We then use Theorem 1 to analyze the expected number of iterations (denoted by $\mathbb{E}[T]$) until finding the Pareto front when starting from any set of Pareto optimal solutions.

Let $\xi \in \mathcal{X}$ model the SEMO solving mCOCZ and $c = 1$. We design the function h by setting $\boldsymbol{\alpha} = \mathbf{0}$, i.e., $h(x) = |x \cap \mathcal{S}^*|$. It is easy to verify that h is well-defined and $r = |F^*| = (n'+1)^{m'}$. For the chain $\xi' \in \mathcal{Y} = \{0,1\}^r$ with $\mathcal{Y}^* = \{1^r\}$, we construct the transition behavior as follows: for any $\xi'_t \notin \mathcal{Y}^*$, ξ'_{t+1} becomes 1^r with probability $1/(|\xi'_t|_0)^{1-1/m'}$; otherwise, $\xi'_{t+1} = \xi'_t$. Thus, $\mathbb{E}[\![\tau' | \xi'_t]\!] = (|\xi'_t|_0)^{1-1/m'}$.

We then show that Eq. (1) holds with $\delta \leq 1 - \frac{1}{2nr}$. For any $\xi_t = x \notin \mathcal{X}^*$, assume that $h(x) = k < r$. Note that for any t, ξ_t contains only incomparable Pareto optimal solutions. Since the number of Pareto optimal solutions in the population never decreases, $h(\xi_{t+1}) \geq h(\xi_t) = k$. We then analyze $\mathrm{P}_g := \mathrm{P}(h(\xi_{t+1}) = k+1)$. For $i \in \{1, \dots, m'\}$, let $A_i = \{f_{2i-1}(\boldsymbol{s}) | \boldsymbol{s} \in x\}$ and $a_i = |A_i|$. Note that $1 \leq a_i \leq n'+1$ and f_{2i-1} corresponds to the i-th block in the second half of a solution, i.e., the bits in positions $\frac{n}{2}+(i-1)\cdot n'+1 \sim \frac{n}{2}+i\cdot n'$. Assume that $a_1 \geq a_2 \geq \dots \geq a_{m'}$. We consider two cases:
(1) $a_{m'} \leq n'$. For the m'-th block, there is at least one missing $f_{2m'-1}$ value. Thus, for any value of f_1 in A_1, there is at least one corresponding solution $\boldsymbol{s}$ in x, and flipping a proper bit in the m'-th block of $\boldsymbol{s}$ can generate a Pareto optimal solution with a new value of $f_{2m'-1}$; this implies that $h(\xi_{t+1}) = k+1$. Since $|x| = k$, $\mathrm{P}_g \geq \frac{a_1}{nk} \geq k^{1/m'}/(nk)$.
(2) $a_{m'} = n'+1$. Let $NF^* = F^* \setminus \{f(\boldsymbol{s}) \mid \boldsymbol{s} \in x\}$ denote the set of objective vectors in F^* that have not been found. For $1 \leq i \leq m'$, let $B_i = \{f^*_{2i-1} \mid (f^*_1, \dots, f^*_m) \in NF^*\}$ and $b_i = |B_i|$. Suppose $j \in \arg\max_i b_i$. For any value $z \in B_j$, there is at least one corresponding solution $\boldsymbol{s} \in x$ with $f_{2j-1}(\boldsymbol{s}) = z$, and flipping a proper bit of $\boldsymbol{s}$ can generate a Pareto optimal solution $\boldsymbol{s'}$ with $f(\boldsymbol{s'}) \in NF^*$; otherwise, all the objective vectors in F^* with $f^*_{2j-1} = z$ have been found, which contradicts with $z \in B_j$. Thus, $\mathrm{P}_g \geq \frac{b_j}{nk} \geq (r-k)^{1/m'}/(nk)$.

Combining these two cases, $\mathrm{P}_g \geq \min\{\frac{k^{1/m'}}{nk}, \frac{(r-k)^{1/m'}}{nk}\}$. By using the same analysis as Eqs. (2-3), Eq. (1) holds with

$$\begin{aligned}\delta &= \mathrm{P}_g \cdot \mathbb{E}[\![\tau' | \xi'_0 = 1^{k+1}0^{r-k-1}]\!] + (1-\mathrm{P}_g)\mathbb{E}[\![\tau' | \xi'_0 = 1^k 0^{r-k}]\!] \\ &\quad - (1 - 1/(r-k)^{1-1/m'}) \cdot \mathbb{E}[\![\tau' | \xi'_1 = 1^k 0^{r-k}]\!] \\ &= \mathrm{P}_g \cdot (r-k)^{1-1/m'}((1 - 1/(r-k))^{1-1/m'} - 1) + 1 \\ &\leq -(\mathrm{P}_g/2) \cdot (r-k)^{-1/m'} + 1,\end{aligned}$$

where the inequality is by $(1-\frac{1}{r-k})^{1-1/m'} \leq 1 - \frac{1-1/m'}{r-k}$ and $m' = \frac{m}{2} \geq 2$. If $k \leq \frac{r}{2}$, $\mathrm{P}_g \geq \frac{k^{1/m'}}{nk} \geq \frac{1}{nr^{1-1/m'}}$, thus $\delta \leq 1 - \frac{1}{2nr}$. If $k > \frac{r}{2}$, $\mathrm{P}_g \geq \frac{(r-k)^{1/m'}}{nk} \geq \frac{(r-k)^{1/m'}}{nr}$, thus $\delta \leq 1 - \frac{1}{2nr}$.

By applying $\delta \leq 1 - \frac{1}{2nr}$ and $\mathbb{E}[\![\tau' | \xi'_0 = 0^r]\!] \leq r^{1-1/m'}$ to Theorem 1, we get that $\mathbb{E}[T] \leq r^{1-1/m'} \cdot 2nr = O(n^m)$. Note that $r = (n'+1)^{m'}$ and $2m' = m$. Thus, the expected running time for $m > 4$ is $O(n^{m/2+1} \log n + n^m) = O(n^m)$.

We further derive a tighter upper bound for $m = 4$ by using a different $\xi' \in \mathcal{Y}$ and analyzing P_g more carefully. The transition behavior of ξ' is as follows: ξ'_{t+1} is generated by flipping the first 0-bit of ξ'_t with probability $|\xi'_0|/r$; otherwise, $\xi'_{t+1} = \xi'_t$. Thus, $\mathbb{E}[\![\tau' | \xi'_t]\!] = rH_{|\xi'_t|_0}$. We then show that Eq. (1) holds with $\delta \leq 1 - \frac{1}{n}$ by re-analyzing P_g. Note that $x \subseteq \mathcal{S}^*$. For any $z \in A_2$, let $C_z \in \arg\max_{C \subseteq x} |C|$ s.t. $\forall \boldsymbol{s} \in C$, $f_3(\boldsymbol{s}) = z$ and $\max_{\boldsymbol{s} \in C} f_1(\boldsymbol{s}) - \min_{\boldsymbol{s} \in C} f_1(\boldsymbol{s}) = |C| - 1$. Let $\boldsymbol{u} = \arg\max_{\boldsymbol{s} \in C_z} f_1(\boldsymbol{s})$ and $\boldsymbol{v} = \arg\min_{\boldsymbol{s} \in C_z} f_1(\boldsymbol{s})$. Then, a new Pareto optimal solution can be generated by flipping one 0-bit in the first block of $\boldsymbol{u}$ or flipping one 1-bit in the first block of $\boldsymbol{v}$, whose probability is $\frac{n' - (f_1(\boldsymbol{u}) - n/2) + f_1(\boldsymbol{v}) - n/2}{nk} = \frac{n'+1-|C_z|}{nk}$. By summing over all $z \in A_2$, we get that one part of P_g is at least $\frac{(n'+1)a_2 - k}{nk}$. By considering $z \in A_1$ and exchanging f_1 and f_3 in the analysis above, we can similarly get that the other part of P_g is at least $\frac{(n'+1)a_1 - k}{nk}$. Thus, $\mathrm{P}_g \geq \frac{(n'+1)(a_1+a_2) - 2k}{nk} \geq \frac{2(n'+1)\sqrt{k} - 2k}{nk}$, where the second inequality is by $a_1 a_2 \geq k$. Then, the δ in Eq. (1) satisfies that

$$\delta = \mathrm{P}_g \cdot \left(-\frac{r}{r-k}\right) + 1 \leq 1 - \frac{2r}{n\sqrt{k}(n'+1+\sqrt{k})} \leq 1 - \frac{1}{n},$$

where the last inequality is by $k < r = (n'+1)^2$. By applying $\delta \leq 1 - \frac{1}{n}$ and $\mathbb{E}[\![\tau' | \xi'_0 = 0^r]\!] = rH_r$ to Theorem 1, we get that $\mathbb{E}[T] = O(n^3 \log n)$. Thus, the expected running time for $m = 4$ is $O(n^3 \log n + n^3 \log n) = O(n^3 \log n)$. □

6 Approximate Analysis

In this section, we consider a variant of the GSEMO solving the `WOMM` problem, a generalization of the previously studied `OneMinMax` problem [Giel and Lehre, 2010; Osuna *et al.*, 2017]. As presented in Definition 7, it is to maximize the weighted sum of 1-bits and 0-bits of a solution at the same time. Note that optimizing two linear functions simultaneously often appears in multi-objective combinatorial problems such as multi-objective minimum spanning trees and multi-objective knapsacks. The Pareto front F^* can be exponentially large, e.g., if $w_i = 2^i$, $|F^*| = 2^n$; thus we analyze the running time until finding an approximation of F^* (i.e., approximate analysis), instead of that until finding F^* (i.e., exact analysis) in the previous two sections. Theorem 5 shows that a good approximation can be obtained in polynomial time, which is consistent with the good empirical performance of MOEAs in solving multi-objective combinatorial problems [Zhou and Gen, 1999; Ishibuchi *et al.*, 2015].

Definition 7 (`WOMM`)**.** *The pseudo-Boolean function WOMM: $\{0,1\}^n \to \mathbb{R}^2$ is defined as*

$$\textit{WOMM}(\boldsymbol{s}) = \left(\textstyle\sum_{i=1}^{n} w_i s_i, \sum_{i=1}^{n} w_i(1 - s_i)\right).$$

where $0 < w_1 \leq w_2 \leq \ldots \leq w_n$.

We introduce two useful techniques into the GSEMO, i.e., the initialization strategy [Qian *et al.*, 2013] and the diversity-selecting mechanism [Horoba, 2009]. The former uses the (1+1)-EA (a simple single-objective EA) to optimize each objective separately, and then uses the optimum of each objective as the initial population. The latter is used to keep a diverse population and then achieve a good spread over the Pareto front. The objective space is divided into boxes, and the box value of a solution $\boldsymbol{s}$ is defined as

$$b_l(\boldsymbol{s}) := \left(\lceil \log_l(1 + f_1(\boldsymbol{s})/w_1) \rceil, \lceil \log_l(1 + f_2(\boldsymbol{s})/w_1) \rceil\right),$$

where $l > 1$. In each iteration, a non-empty box (i.e., a box which contains at least one solution in the population) is chosen uniformly at random (u.a.r.); then a solution in the chosen box is selected u.a.r. to generate a new solution. Let $W = \sum_{i=1}^{n} w_i$ for simplicity. Lemma 2 gives an upper bound on the number of non-empty boxes.

Lemma 2. *For the GSEMO on WOMM, there are at most $B = 2(\log_l(1 + W/w_1) + 2)$ non-empty boxes in the population.*

Proof. Assume that in the population, there are $m > B$ different non-empty boxes: $(a_1, b_1), (a_2, b_2), \ldots, (a_m, b_m)$, where $a_1 \leq a_2 \leq \ldots \leq a_m$, and for $i < j$, $b_i > b_j$ if $a_i = a_j$. Then, it must hold that $b_1 \geq b_2 \geq \ldots \geq b_m$; otherwise, it implies that for some i, $a_i < a_{i+1}$ and $b_i < b_{i+1}$, which contradicts with the fact that the solutions in the population are incomparable. Thus, we get $a_1 - b_1 \leq a_2 - b_2 \leq \ldots \leq a_m - b_m$. Note that $a_i, b_i \in \{0, 1, \ldots, L\}$, where $L = \lceil \log_l(1 + W/w_1) \rceil$, and there are at most $2L + 1$ different values for $a_i - b_i$. Meanwhile, $m > B > 2L + 2$. Thus, for some i, $a_i - b_i = a_{i+1} - b_{i+1}$; this implies that $a_i = a_{i+1}$, $b_i = b_{i+1}$, which contradicts with the assumption. □

Lemma 3. $A = \{0^n, 10^{n-1}, 010^{n-2}, \ldots, 0^{n-1}1\}$ *is a $\frac{1}{n}$-approximation of F^*. Furthermore, the approximation ratio is $\max\{\frac{\delta}{1+\delta}, \frac{1}{n}\}$, if $\forall i, w_{i+1} \geq (1+\delta) \cdot w_i$, where $\delta > 0$.*

Proof. We only need to show that for any $\boldsymbol{s} \in \{0,1\}^n$, there exists a solution $\boldsymbol{s'} \in A$ such that $\forall i \in \{1, 2\} : f_i(\boldsymbol{s'}) \geq f_i(\boldsymbol{s})/n$. Let $k = \max\{i | s_i = 1\}$. We consider two cases: (1) If $k = 0$, $\boldsymbol{s} = 0^n \in A$. (2) If $1 \leq k \leq n$, $f_2(0^{k-1}10^{n-k}) = W - w_k \geq f_2(\boldsymbol{s})$; $f_1(0^{k-1}10^{n-k})/f_1(\boldsymbol{s}) \geq w_k/(\sum_{i=1}^{k} w_i) \geq 1/n$, since w_i increases with i. For the furthermore clause, $(\sum_{i=1}^{k} w_i)/w_k \leq \frac{1-1/(1+\delta)^k}{1-1/(1+\delta)} \leq \frac{1+\delta}{\delta}$, since $w_{i+1} \geq (1+\delta) \cdot w_i$; then, $w_k/(\sum_{i=1}^{k} w_i) \geq \delta/(1+\delta)$. Thus, the lemma holds. □

Theorem 5. *For maximizing WOMM, the GSEMO with an initialization strategy and a diversity-selecting mechanism can find a $\frac{1}{n}$-approximation of the Pareto front in $O(n^2(\log_l n + \log_l(w_n/w_1)))$ expected running time, where $l > 1$. Furthermore, the approximation ratio is $\max\{\frac{\delta}{1+\delta}, \frac{1}{n}\}$, if $\forall i, w_{i+1} \geq (1+\delta) \cdot w_i$, where $\delta > 0$.*

Proof. Since the (1+1)-EA optimizes a linear function in $O(n \log n)$ expected time [Droste *et al.*, 2002], the initialization costs $O(n \log n)$ time, and the population will contain 1^n and 0^n. We then use Theorem 1 to analyze the expected number of iterations until finding a $\frac{1}{n}$-approximation of F^*.

Let $\xi \in \mathcal{X}$ model the variant of the GSEMO solving `WOMM` and $c = \frac{1}{n}$. We design a well-defined function h by setting $\boldsymbol{\alpha} = \boldsymbol{0}$, i.e., $h(x) = g(F^*, x, \frac{1}{n})$. Note that $r = |F^*|$. For the chain $\xi' \in \mathcal{Y} = \{0,1\}^r$ with $\mathcal{Y}^* = \{1^r\}$, its transition behavior is constructed as follows: for any $\xi'_t \notin \mathcal{Y}^*$, ξ'_{t+1} becomes 1^r with probability $1/(1 + \log |\xi'_t|_0)$; otherwise, $\xi'_{t+1} = \xi'_t$. We can easily derive that $\mathbb{E}[\![\tau' | \xi'_t]\!] = 1 + \log |\xi'_t|_0$ for $\xi'_t \notin \mathcal{Y}^*$.

Then, we are to investigate Eq. (1). For any $\xi_t = x \notin \mathcal{X}^*$, assume $h(x) = k < r$. It is easy to verify that $h(\xi_{t+1}) \geq h(\xi_t) = k$. Note that the solution 0^n will be always contained in the population, since no other solution can weakly dominate it. Furthermore, it has a unique box value $b_l(0^n) = (0, \lceil \log_l(1 + W/w_1) \rceil)$, since for any $\boldsymbol{s} \neq 0^n$, $\lceil \log_l(1 + f_1(\boldsymbol{s})/w_1) \rceil \geq 1$. Assume that there are j solutions (denoted by $\boldsymbol{s^1}, \boldsymbol{s^2}, \ldots, \boldsymbol{s^j}$) in A which are not weakly dominated by any solution in ξ_t. Note that $j \geq 1$; otherwise, $h(\xi_t) = r$, which makes a contradiction. For $1 \leq i \leq j$, $\boldsymbol{s^i}$ can be generated by choosing 0^n from ξ_t and flipping a specific 0-bit, whose probability is at least $\frac{1}{B} \cdot \frac{1}{n}(1 - \frac{1}{n})^{n-1} \geq \frac{1}{Ben}$. Once $\boldsymbol{s^i}$ is generated, it will be added into the population, since no solution in ξ_t weakly dominates it. For $1 \leq i \leq j$, let $k_i = h(x \cup \{\boldsymbol{s^i}\})$, then $k_i \geq k$. By using the same analysis as Eqs. (2) and (3), we have

$$\begin{aligned}
\delta &\leq (1/(Ben)) \cdot \textstyle\sum_{i=1}^{j} \mathbb{E}[\![\tau' | \xi'_0 = 1^{k_i}0^{r-k_i}]\!] \\
&\quad + (1 - j/(Ben)) \cdot \mathbb{E}[\![\tau' | \xi'_0 = 1^k 0^{r-k}]\!] \\
&\quad - (1 - 1/(1 + \log(r - k))) \cdot \mathbb{E}[\![\tau' | \xi'_1 = 1^k 0^{r-k}]\!] \\
&= 1 + (1/(Ben)) \cdot \textstyle\sum_{i=1}^{j} \log\left((r - k_i)/(r - k)\right) \\
&\leq 1 + (1/(Ben)) \cdot \textstyle\sum_{i=1}^{j} (k - k_i)/(r - k) \leq 1 - 1/(Ben),
\end{aligned}$$

where the second inequality is by $\log t \leq t - 1$ for $t > 0$, and the last is by $\sum_{i=1}^{j}(k_i - k) = \sum_{i=1}^{j}(h(x \cup \{\boldsymbol{s^i}\}) - h(x)) \geq h(x \cup A) - h(x) = r - k$. Thus, Eq. (1) holds with $\delta \leq 1 - 1/(Ben)$.

By applying $\delta \leq 1 - \frac{1}{Ben}$ and $\mathbb{E}[\![\tau' | \xi'_0 = 0^r]\!] \leq 1 + n \log 2$ to Theorem 1, we get that the expected running time for finding a $\frac{1}{n}$-approximation of F^* is at most $(1 + n \log 2) Ben = O(n^2(\log_l n + \log_l (w_n/w_1)))$. For the furthermore clause, the analysis still holds by setting $c = \max\{\frac{\delta}{1+\delta}, \frac{1}{n}\}$. □

7 Conclusion

In this paper, we propose a general approach for deriving upper bounds on the running time of MOEAs. The key is to reduce the analysis of a given MOEA process to that of an easy-to-analyze process. We apply this approach to diverse situations, including bi-objective and many-objective optimization as well as exact and approximate analysis. Our analysis can give the leading constants as well as bring an asymptotic improvement for some known running time bounds, which displays the strength of the proposed approach. In addition, our results provide some theoretical justification for MOEAs well solving multi-objective combinatorial problems in practice.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China (2017YFC0804002), the NSFC (61603367, 61672478), the YESS (2016QNRC001), and the Science and Technology Innovation Committee Foundation of Shenzhen (ZDSYS201703031748284).

References

[Auger and Doerr, 2011] A. Auger and B. Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific, Singapore, 2011.

[Bäck, 1996] T. Bäck. *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*. Oxford University Press, Oxford, UK, 1996.

[Coello Coello and Lamont, 2004] C. A. Coello Coello and G. B. Lamont. *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, Singapore, 2004.

[Doerr *et al.*, 2013] B. Doerr, B. Kodric, and M. Voigt. Lower bounds for the runtime of a global multi-objective evolutionary algorithm. In *CEC*, pages 432–439, 2013.

[Doerr *et al.*, 2016] B. Doerr, W. Gao, and F. Neumann. Runtime analysis of evolutionary diversity maximization for ONEMINMAX. In *GECCO*, pages 557–564, 2016.

[Droste *et al.*, 2002] S. Droste, T. Jansen, and I. Wegener. On the analysis of the (1+1) evolutionary algorithm. *Theoretical Computer Science*, 276(1-2):51–81, 2002.

[Friedrich *et al.*, 2010] T. Friedrich, N. Hebbinghaus, and F. Neumann. Plateaus can be harder in multi-objective optimization. *Theoretical Computer Science*, 411(6):854–864, 2010.

[Friedrich *et al.*, 2011] T. Friedrich, C. Horoba, and F. Neumann. Illustration of fairness in evolutionary multi-objective optimization. *Theoretical Computer Science*, 412(17):1546–1556, 2011.

[Giel and Lehre, 2010] O. Giel and P. K. Lehre. On the effect of populations in evolutionary multi-objective optimisation. *Evolutionary Computation*, 18(3):335–356, 2010.

[Giel, 2003] O. Giel. Expected runtimes of a simple multi-objective evolutionary algorithm. In *CEC*, pages 1918–1925, 2003.

[Horoba, 2009] C. Horoba. Analysis of a simple evolutionary algorithm for the multiobjective shortest path problem. In *FOGA*, pages 113–120, 2009.

[Ishibuchi *et al.*, 2015] H. Ishibuchi, N. Akedo, and Y. Nojima. Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems. *IEEE Transactions on Evolutionary Computation*, 19(2):264–283, 2015.

[Laumanns *et al.*, 2004] M. Laumanns, L. Thiele, and E. Zitzler. Running time analysis of multiobjective evolutionary algorithms on pseudo-Boolean functions. *IEEE Transactions on Evolutionary Computation*, 8(2):170–182, 2004.

[Neumann and Theile, 2010] F. Neumann and M. Theile. How crossover speeds up evolutionary algorithms for the multi-criteria all-pairs-shortest-path problem. In *PPSN*, pages 667–676, 2010.

[Neumann and Witt, 2010] F. Neumann and C. Witt. *Bioinspired Computation in Combinatorial Optimization: Algorithms and Their Computational Complexity*. Springer-Verlag, Berlin, Germany, 2010.

[Neumann, 2007] F. Neumann. Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *European Journal of Operational Research*, 181(3):1620–1629, 2007.

[Neumann, 2012] F. Neumann. Computational complexity analysis of multi-objective genetic programming. In *GECCO*, pages 799–806, 2012.

[Osuna *et al.*, 2017] E. C. Osuna, W. Gao, F. Neumann, and D. Sudholt. Speeding up evolutionary multi-objective optimisation through diversity-based parent selection. In *GECCO*, pages 553–560, 2017.

[Qian *et al.*, 2013] C. Qian, Y. Yu, and Z.-H. Zhou. An analysis on recombination in multi-objective evolutionary optimization. *Artificial Intelligence*, 204:99–119, 2013.

[Qian *et al.*, 2016] C. Qian, K. Tang, and Z.-H. Zhou. Selection hyper-heuristics can provably be helpful in evolutionary multi-objective optimization. In *PPSN*, pages 835–846, 2016.

[Yu *et al.*, 2015] Y. Yu, C. Qian, and Z.-H. Zhou. Switch analysis for running time analysis of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 19(6):777–792, 2015.

[Zhou and Gen, 1999] G. Zhou and M. Gen. Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research*, 114(1):141–152, 1999.