



# On Performance of a Simple Multi-objective Evolutionary Algorithm on the Constrained Minimum Spanning Tree Problem

Xinsheng Lai<sup>1</sup>

Received: 18 December 2021 / Accepted: 19 July 2022  
© The Author(s) 2022

## Abstract

The constrained optimization problems can be transformed into multi-objective optimization problems, and thus can be optimized by multi-objective evolutionary algorithms. This method has been successfully used to solve the constrained optimization problems. However, little theoretical work has been done on the performance of multi-objective evolutionary algorithms for the constrained optimization problems. In this paper, we theoretically analyze the performance of a multi-objective evolutionary algorithms on the constrained minimum spanning tree problem. First, we theoretically prove that the multi-objective evolutionary algorithm is capable of finding a (2,1)-approximation solution for the constrained minimum spanning tree problem in a pseudopolynomial runtime. Then, this simple multi-objective evolutionary algorithm is shown to be efficient on a constructed instance of the problem.

**Keywords** Approximation algorithm · Multi-objective evolutionary algorithm · Constrained optimization · Spanning tree problem · Combinatorial problem

## Abbreviations

CMST	Constrained minimum spanning tree
EAs	Evolutionary algorithms
MOEAs	Multi-objective evolutionary algorithms
GSEMO	Simple multi-objective evolutionary algorithm that searches globally
BMST	Biobjective minimum spanning tree

## 1 Introduction

The constrained minimum spanning tree (CMST) problem aims to find a spanning tree in a connected undirected graph where each edge has two costs: weight and length, respectively, such that the total weight is minimized under the condition that the total length is not larger than a given budget  $L$ . Since it is NP-hard [1], researchers tend to believe that there is no polynomial time algorithm for solving this problem, and devote to exploring polynomial time approximation schemes for the CMST problem [2–6].

As general-purpose and powerful search engines, evolutionary algorithms (EAs) have been extensively used to solve constrained optimization problems [7–14]. There are several constraint-handling methods used with EAs. The multi-objective optimization method is among the most popular ones [15–18]. This method first transforms a constrained optimization problem into a multi-objective optimization problem. Then, multi-objective EAs (MOEAs) or other multi-objective optimization algorithms are used to solve the resulting multi-objective optimization problem. At last, the solutions found by multi-objective optimization algorithms are transformed back into the original constrained optimization problem.

Recently, the theoretical analysis of the approximation performance of EAs on NP-hard combinatorial optimization problems has attracted more and more attentions from researchers. One reason for this phenomenon may be that a large number of combinatorial optimization problems are NP-hard, for which we tend to believe that there is no algorithms being able to find their globally optimal solutions in polynomial runtime. Thus, we are interested in what approximation solutions EAs can find in polynomial runtime on such problems. Another reason may be that the theoretical analysis of the approximation performance of EAs on

✉ Xinsheng Lai  
xsl2001\_jx@163.com

<sup>1</sup> Department of Computer Science and Engineering,  
Shaoxing University, Beihai Street, Shaoxing 312000,  
Zhejiang, China

NP-hard combinatorial optimization problems is helpful in deeply understanding the mechanism of EAs.

The approximation performance of the (1+1) EA has been studied on some single objective combinatorial optimization problems [19–21], and the MOEA's approximation performance has been analyzed on some single objective [21–23] and some multi-objective combinatorial optimization problems [24, 25] in recent years.

For constrained optimization problems, Zhou and He [26] analyzed the time complexity of the (1+1) EA with penalty function methods for solving constrained optimization problems. They showed that higher penalty coefficients may be good choices for some instances of constrained optimization problems, while on other instances lower penalty coefficients may be good choices. Qian et al. [27] theoretically compared EAs using penalty function methods with EAs using multi-objective optimization methods on two classes of constrained optimization problems: the minimum matroid optimization problem (P-solvable) and the minimum cost coverage problem (NP-hard). They found that EAs using multi-objective optimization method are more efficient to find the optimal and approximation solutions, respectively.

The CMST belongs to the class of constrained optimization problems, which are still attractive to researchers [28–30]. There is still no theoretical investigation on the approximation performance of the EA on this problem. This paper aims to discover what solutions to the CMST problem can be found by a simple MOEA within a polynomial runtime. This simple MOEA is identical to the simple multi-objective evolutionary algorithm that searches globally (GSEMO) in [24]. To use GSEMO to solve the CMST problem, we should first transform the CMST problem into a biobjective minimum spanning tree (BMST) problem by considering the total length of a minimum spanning tree as an additional objective. Then, we focus on the performance of GSEMO on this resulting BMST problem.

The contributions of this paper are as follows. First, it is revealed that GSEMO can find a (2,1)-approximation solution for the CMST problem in a pseudopolynomial runtime. Second, it is illustrated that the MOEA is capable of finding the optimal solutions for some instances of the CMST problem.

The remainder of this paper is organized as follows. Next section describes the constrained minimum spanning tree problem and GSEMO. Section 3 reveals the approximation performance of GSEMO on the CMST problem. Section 4 constructs an instance of the CMST problem and analyzes the performance of GSEMO on it. Finally, the last section concludes this paper.

## 2 The Constrained Minimum Spanning Tree Problem and GSEMO

In this section, we describe the CMST problem and GSEMO. At the beginning, we define some concepts to be used in this paper.

### 2.1 Related Concepts

The first concept is the concept of a spanning subgraph.

**Definition 1** (*spanning subgraph*) Given two graphs  $G = (V, E)$  and  $G' = (V', E')$ , where  $V(V')$  and  $E(E')$  are the sets of nodes and edges of  $G(G')$ , respectively, if  $V' = V$  and  $E' \subseteq E$ , then  $G'$  is a spanning subgraph of  $G$ .

Clearly, a spanning subgraph may be unconnected, and may contain cycles. We then define a spanning tree based on the concept of a spanning subgraph.

**Definition 2** (*spanning tree*) Given two connected undirected graphs  $G = (V, E)$  and  $G' = (V', E')$ , where  $V(V')$  and  $E(E')$  are the sets of nodes and edges of  $G(G')$ , respectively, if  $G'$  is a connected spanning subgraph of  $G$  and the number of edges contained in  $E'$  equals  $|V| - 1$ , where  $|V|$  is the number of nodes in  $G$ , then  $G'$  is a spanning tree of  $G$ .

The last one is the relationship “ $\leq$ ” between two 2-dimensional vectors.

**Definition 3** For two 2-dimensional vectors  $u = (u_1, u_2)$  and  $v = (v_1, v_2)$ , if one of the following two conditions is satisfied:

- (1)  $u_1 < v_1$  and  $u_2 \leq v_2$ ;
- (2)  $u_1 \leq v_1$  and  $u_2 < v_2$ ,

then we denote it by  $u \leq v$ .

### 2.2 The Constrained Minimum Spanning Tree Problem

In a given connected undirected graph  $G = (V, E)$ , where  $V$  and  $E$  are the sets of  $n$  nodes and  $m$  edges, respectively, each edge  $e \in E$  associates with two different nonnegative costs  $w_e$  and  $l_e$ . We denote the set of edges contained in a spanning tree  $T$  of  $G$  by  $E(T)$ . The CMST problem is to find a spanning tree  $T$  of  $G$ , such that  $\sum_{e \in E(T)} l_e \leq L$  and  $\sum_{e \in E(T)} w_e$  is minimized, where  $L$  is a given budget. We refer to  $w_e$  and  $l_e$  as the weight and the length of  $e$ , respectively.

The CMST problem can be formulated as follows:

$$\begin{aligned}
 \min \quad & f_1(T) := \sum_{e \in E(T)} w_e \\
 \text{s.t.} \quad & f_2(T) := \sum_{e \in E(T)} l_e \leq L, \\
 & T \in \mathcal{T},
 \end{aligned} \tag{1}$$

where  $\mathcal{T}$  is the set of all spanning trees of  $G$ .

Though the CMST problem has only one objective function to be optimized, with a constraint, it becomes an NP-hard problem, which we usually think that there is no algorithm being able to find its globally optimal solution in a polynomial runtime.

If an algorithm can obtain a spanning tree  $T$  for the CMST problem in a polynomial runtime with  $l(T) \leq \alpha L$  and  $w(T) \leq \beta W$ , where  $W$  is the minimum weight among all spanning trees of the input graph whose lengths are no more than  $L$ , then we call this algorithm an  $(\alpha, \beta)$ -approximation algorithm for the CMST problem.

### 2.3 GSEMO

If the CMST problem is transformed into a BMST problem, then the MOEA can be used to find solutions for the transformed BMST problem. Let  $S$  be the set of all possible selections of the edges of  $G$ . Since each selection corresponds a subset of the edge set  $E$ , each selection determines one and only one spanning subgraph of  $G$ . Therefore,  $S$  is the set of all spanning subgraphs of  $G$ . Obviously,  $\mathcal{T} \subseteq S$ .

GSEMO will search in  $S$ . We call  $S$  the search space and call a possible selection in  $S$  a solution. Denote the number of edges contained in  $G$  by  $m$ , and describe all edges in a fixed sequence  $(e_1, e_2, \dots, e_m)$ . We encode a solution  $X$  as a bit string  $(x_1, x_2, \dots, x_m) \in \{0, 1\}^m$ , where  $x_i = 1$  if edge  $e_i$  is selected and  $x_i = 0$  otherwise. Clearly, each bit string in  $\{0, 1\}^m$  corresponds to one solution in  $S$ , and vice versa. Thus, bit string and solution can be interchangeably used.

For solution  $X \in S$ , let  $H(X)$  denote the spanning subgraph corresponds to  $X$ , and  $|X|$  the number of edges contained in  $H(X)$ ,  $w(X)$  ( $l(X)$ ) the sum of weight (length) of each edge contained in  $H(X)$ , i.e.,  $|X| = \sum_{i=1}^m x_i$ ,  $w(X) = \sum_{i|x_i=1} w(e_i)$ , and  $l(X) = \sum_{i|x_i=1} l(e_i)$ . Let  $u_{wl} = \max\{w_{max}, l_{max}\}$ ,  $w_u = n^2 \cdot u_{wl}$ , and  $l_{wl} = \min\{w_{max}, l_{max}\}$ , where  $w_{max}$  and  $l_{max}$  are the maximum values of weight and length on all edges of  $G$ , respectively.

The fitness function of any solution  $X \in S$  is commonly defined as follows [31]:

$$\begin{aligned}
 \bar{f}(X) &= (\bar{f}_1(X), \bar{f}_2(X)), \\
 \bar{f}_1(X) &= (c(X) - 1) \cdot w_u^2 + (|X| - (n - 1)) \cdot w_u + w(X),
 \end{aligned}$$

where  $c(X)$  denotes the number of connected components in  $H(X)$ .

For solution  $X$ , if  $H(X)$  is unconnected, then the fitness function ensures the decrease of the number  $c(X)$  of connected components of  $H(X)$ ; if  $H(X)$  is already connected, the fitness function ensures the decrease of the number of redundant edges contained in  $H(X)$  under the condition of keeping  $H(X)$  a connected spanning subgraph unchanged; if  $H(X)$  is a spanning tree, i.e.,  $X \in \mathcal{T}$ , then  $\bar{f}_1(X)$  and  $\bar{f}_2(X)$  of the fitness function  $\bar{f}(X)$  are the weight and the length of a spanning tree, respectively, i.e.,  $\bar{f}_1(X) = f_1(X)$  and  $\bar{f}_2(X) = f_2(X)$ . For two solutions  $X_1, X_2 \in S$ , if  $f(X_1) \leq f(X_2)$ , then we say that  $X_1$  dominates  $X_2$ .

Let  $S_u$  and  $S_c$  denote the subspaces of  $S$  including all unconnected spanning subgraphs and all connected spanning subgraphs, respectively. Clearly, the fitness function can guide the search of GSEMO from  $S_u$  into  $S_c$ , and finally into  $\mathcal{T}$ .

GSEMO starts with a population containing only a solution selected from  $S$  uniformly at random. If the termination criterion is not fulfilled, then a solution is selected from the population uniformly at random. An offspring solution is obtained by flipping each bit of the selected solution with probability  $1/m$ . If this offspring solution cannot be dominated by any solution in the population, then it is included in the population; meanwhile, all solutions dominated by this offspring solution are removed from the population. GSEMO for the transformed BMST problem can be described as Algorithm 1.

#### Algorithm 1: GSEMO

- 01: **Begin**
- 02: Initialize a solution  $X \in \{0, 1\}^m$  uniformly at random;
- 03:  $P \leftarrow \{X\}$ ;
- 04: **While** termination criterion is not fulfilled
- 05:     Choose a solution  $X$  from  $P$  uniformly at random;
- 06:     Obtain an offspring  $Y$  by flipping each bit in  $X$  with probability  $1/m$ ;
- 07:     **If**  $\forall X \in P: Y$  is not dominated by  $X$  and  $f(Y) \neq f(X)$  **then**
- 08:          $Q := \{X | X \in P, Y \text{ dominates } X\}$ ;
- 09:          $P \leftarrow P \cup \{Y\} \setminus Q$ ;
- 10:     **End while**
- 11: **End**

The termination criterion of GSEMO may be a solution with a certain quality having been found or a given number of iterations having been reached. We care about how many iterations GSEMO should run to find a (set of) solution(s) with given quality. Therefore, the runtime in this paper refers to the number of iterations that GSEMO runs.

### 3 The Approximation Performance of GSEMO on the CMST Problem

If the weight and length on each edge are positive integer numbers, then we will reveal that GSEMO include in its population a (2,1)-approximation solution for the CMST problem in a pseudopolynomial runtime.

First, we prove that among all spanning trees with objective vectors on the convex hull of the Pareto front of the transformed BMST problem, there is a (2,1)-approximation solution to the CMST problem. Then, we reveal that GSEMO can include this approximation solution in its population in pseudopolynomial runtime.

To prove that there is a (2,1)-approximation solution whose objective vector is on the convex hull of the Pareto front of the transformed BMST problem, we should discuss Lagrangian relaxation of the CMST problem, and then discuss the relationship between Lagrangian relaxation of the CMST problem and the transformed BMST problem.

#### 3.1 Lagrangian Relaxation of the CMST Problem

Lagrangian relaxation is generally a method to approach the global optimum for constrained optimization problems [32, 33]. It can also be used to approach the global optimum of the CMST problem [2].

By relaxing the budget constraint  $\sum_{e \in E(T)} l_e \leq L$ , we obtain the Lagrangian relaxation of the CMST problem, which is a minimum spanning tree problem for any  $\bar{\lambda} (\geq 0)$  as described in the following:

$$l(\bar{\lambda}) := \min_{T \in \mathcal{T}} f_1(T) + \bar{\lambda}(f_2(T) - L) \tag{2}$$

We call  $\bar{\lambda}$  Lagrangian multiplier.

If we are able to compute the optimal solution for the relaxed problem (2) with a fixed Lagrangian multiplier  $\bar{\lambda} \geq 0$ , then  $l(\bar{\lambda})$  is a lower bound of the optimal value of the CMST problem.

In fact, let  $T^*$  be an optimal solution of the CMST problem, then we have  $l(\bar{\lambda}) \leq f_1(T^*) + \bar{\lambda}(f_2(T^*) - L)$ . Together with  $f_2(T^*) - L \leq 0$  as  $T^*$  obviously satisfies the budget constraint, we have  $l(\bar{\lambda}) \leq f_1(T^*)$ .

Assume that the optimal value of the problem (1) is  $W_{opt}$ , then  $l(\bar{\lambda}) \leq W_{opt}$ .

To get the best lower bound on  $W_{opt}$ , one should maximize  $l(\bar{\lambda})$  over all  $\bar{\lambda} \geq 0$ . If we denote  $\max_{\bar{\lambda} \geq 0} l(\bar{\lambda})$  by  $LR$ , then  $LR$  is the best lower bound on  $W_{opt}$ . Note that problem (2) on a fixed  $\bar{\lambda}$  is just a minimum spanning tree problem with

respect to the costs  $c_e = w_e + \bar{\lambda}l_e$ . We denote by  $\bar{\lambda}^*$  the value of  $\bar{\lambda}$  that maximizes  $l(\bar{\lambda})$ , and let  $c_e^* = w_e + \bar{\lambda}^* l_e$ .

It is well known that the function  $l(\bar{\lambda})$  is concave and piecewise linear as those thick line segments shown in Fig. 1. In Fig. 1, each spanning tree  $T \in \mathcal{T}$  with weight  $w(T)$  and length  $l(T)$  corresponds to a line function in space  $(l, \bar{\lambda})$  with intercept  $w(T)$  and slope  $l(T) - L$ , and the lower envelope of all these line functions describes the function  $l(\bar{\lambda})$ . We denote all these line functions describing  $l(\bar{\lambda})$  by  $g_i (i \in \{0, 1, \dots, s\})$  with slopes  $k_i$ , such that  $k_i < k_{i+1}$  for all  $i \in \{0, 1, \dots, s\}$ .

#### 3.2 The Transformed Biobjective Minimum Spanning Tree Problem

The CMST problem can be transformed into a BMST problem by considering the total length of a spanning tree as an additional objective as follows:

$$\begin{aligned} \min & f_1(T) \\ \min & f_2(T) \\ \text{s.t.} & T \in \mathcal{T}. \end{aligned} \tag{3}$$

The transformed BMST problem (3) has two objective functions  $f_1$  and  $f_2$ . We denote by  $f$  the vector  $(f_1(T), f_2(T))$ , i.e.,  $f := (f_1(T), f_2(T))$ , and call  $f$  the objective vector.

For two spanning trees  $T_1$  and  $T_2$ , if  $f(T_1) \leq f(T_2)$ , then we say that  $T_1$  dominates  $T_2$ .

**Definition 4 (Pareto optimal solution)** For the transformed BMST problem (3), if a spanning tree  $T$  cannot be dominated by any spanning tree  $T' \in \mathcal{T}$ , then  $T$  is a Pareto optimal solution for the transformed BMST problem (3).

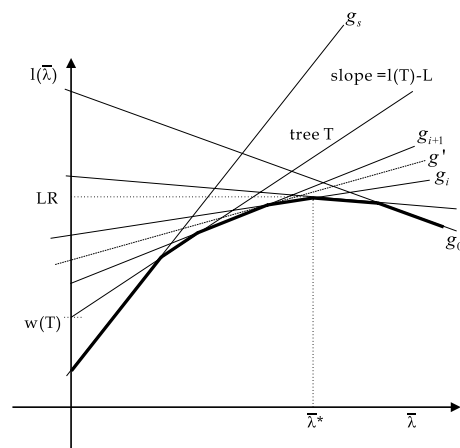


Fig. 1 Function  $l(\bar{\lambda})$  in space  $(l, \bar{\lambda})$

Generally, there are many Pareto optimal solutions for the transformed BMST problem (3).

The set of all Pareto optimal solutions of the transformed BMST problem (3) is called the Pareto set. The objective vector of any Pareto optimal solution is called a Pareto optimal objective vector, and the set of all Pareto optimal objective vectors is called the Pareto front. The convex hull of the Pareto front of the transformed BMST problem consists of line segments which connect two adjacent outermost Pareto optimal objective vectors. Figure 2 shows the Pareto front and the convex hull of the transformed BMST problem (3).

Scalarization approaches can be used to find some Pareto optimal solutions for multi-objective optimization problems, and the weighted-sums approach is the commonly used one [34, 35]. In Fig. 2, all circles including hollow circles and solid circles represent the Pareto optimal objective vectors, and the lines connecting the solid circles describe the convex hull of the Pareto front of problem (3).

Let  $0 \leq \lambda \leq 1$ , then the transformed BMST problem (3) can be formulated by the weighted-sums approach as follows:

$$s(\lambda) = \min \lambda f_1(T) + (1 - \lambda)f_2(T) \quad (4)$$

$$\text{s.t. } T \in \mathcal{T}.$$

According to [35], the weighting-sums approach can only find each Pareto optimal solution with objective vector on the convex hull of the Pareto front of the transformed BMST problem.

Assume that the convex hull of the Pareto front of the transformed BMST problem is described by line functions  $l_1, \dots, l_h$  with slopes  $k'_i < k'_{i+1}$  for all  $i \in \{1, \dots, h\}$ . Let  $p_i$  ( $1 \leq i \leq h - 1$ ) be the intersection point of line functions  $l_i$  and  $l_{i+1}$ . In addition, let  $p_0$  and  $p_h$  be the Pareto optimal objective vectors with minimum value with respect to  $f_1$

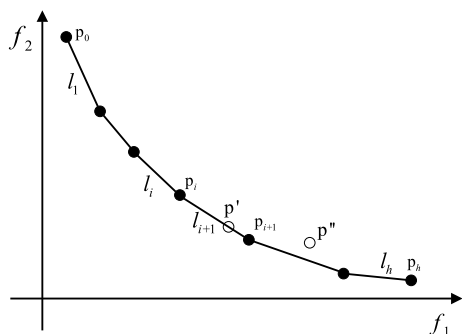


Fig. 2 The Pareto front and the convex hull of the Pareto front of the transformed BMST problem

and  $f_2$ , respectively. Obviously,  $p_0, p_1, \dots, p_h$  are Pareto optimal objective vectors of the transformed BMST problem. We call  $p_0, p_1, \dots, p_h$  extremal points.

Note that not all of Pareto optimal objective vectors are on the convex hull of the transformed BMST problem. For example, some Pareto optimal objective vectors may be above the convex hull, such as the point  $p''$  shown in Fig. 2.

### 3.3 Relationship Between Lagrangian Relaxation of the CMST Problem and the Transformed BMST Problem

There exists an intrinsic relationship between the weighted-sums approach and Lagrangian relaxation [36, 37]. Based on this relationship, we reveal the relationship between Lagrangian relaxation of the CMST problem and the transformed BMST problem in this subsection.

From the forms of problems (2) and (4), we have the following lemma.

**Lemma 1** For the CMST problem, if a spanning tree can be found by Lagrangian relaxation with Lagrangian multiplier  $\bar{\lambda}$ , then it can also be found by the weighted-sums approach with weighting coefficient  $\lambda = 1/(1 + \bar{\lambda})$ . Conversely, if a spanning tree can be found by the weighted-sums approach with weighting coefficient  $\lambda$  and  $\lambda > 0$  ( $= 0$ ), then it can also be found by Lagrangian relaxation with Lagrangian multiplier  $\bar{\lambda} = (1 - \lambda)/\lambda$  ( $= +\infty$ ).

**Proof** For the CMST problem, if a spanning tree  $T'$  can be found by Lagrangian relaxation with a fixed Lagrangian multiplier  $\bar{\lambda}$ , then we have  $f_1(T') + \bar{\lambda}(f_2(T') - L) = \min_{T \in \mathcal{T}} f_1(T) + \bar{\lambda}(f_2(T) - L)$ . Since  $\bar{\lambda}$  is fixed and  $L$  is constant, by adding  $\bar{\lambda}L$  to both sides of the equation, we have  $f_1(T') + \bar{\lambda}f_2(T') = \min_{T \in \mathcal{T}} f_1(T) + \bar{\lambda}f_2(T)$ . Furthermore, we obtain  $\frac{1}{1+\bar{\lambda}}f_1(T') + \frac{\bar{\lambda}}{1+\bar{\lambda}}f_2(T') = \min_{T \in \mathcal{T}} \frac{1}{1+\bar{\lambda}}f_1(T) + \frac{\bar{\lambda}}{1+\bar{\lambda}}f_2(T)$ , since  $\bar{\lambda} \geq 0$ . Let  $\lambda = 1/(1 + \bar{\lambda})$ , and we have  $\lambda f_1(T') + (1 - \lambda)f_2(T') = \min_{T \in \mathcal{T}} \lambda f_1(T) + (1 - \lambda)f_2(T)$ , which means that the solution  $T'$  can also be found by the weighted-sums approach with weighting coefficient  $\lambda = 1/(1 + \bar{\lambda})$ .

If a spanning tree  $T'$  can be found by Lagrangian relaxation with a fixed weighting coefficient  $\lambda > 0$ , then we have  $\lambda f_1(T') + (1 - \lambda)f_2(T') = \min_{T \in \mathcal{T}} \lambda f_1(T) + (1 - \lambda)f_2(T)$ . Since  $\lambda$  is fixed and  $L$  is constant, by adding  $-(1 - \lambda)L$  to both sides of the equation, we have  $\lambda f_1(T') + (1 - \lambda)(f_2(T') - L) = \min_{T \in \mathcal{T}} \lambda f_1(T) + (1 - \lambda)(f_2(T) - L)$ . Furthermore, we obtain  $f_1(T') + \frac{(1-\lambda)}{\lambda}(f_2(T') - L) = \min_{T \in \mathcal{T}} f_1(T) + \frac{(1-\lambda)}{\lambda}(f_2(T) - L)$ , since  $\lambda > 0$ . Let  $\bar{\lambda} = (1 - \lambda)/\lambda$ , we have  $f_1(T') + \bar{\lambda}(f_2(T') - L) = \min_{T \in \mathcal{T}} f_1(T) + \bar{\lambda}(f_2(T) - L)$ ,



which means that the solution  $T'$  can also be found by the Lagrangian relaxation with Lagrangian multiplier  $\bar{\lambda} = (1 - \lambda)/\lambda$ .

If a spanning tree  $T'$  can be found by Lagrangian relaxation with a fixed weighting coefficient  $\lambda = 0$ , then we have  $f_2(T') = \min_{T \in \mathcal{T}} f_2(T)$ , which means that  $T'$  is a minimum spanning tree with respect to  $f_2$ . Thus, it can also be produced by Lagrangian relaxation with a sufficient large Lagrangian coefficient [33], which we denote by  $+\infty$ .

□

### 3.4 Approximation Performance of GSEMO on the CMST Problem

Note that the Lagrangian relaxation of the CMST problem is a minimum spanning tree problem with respect to the costs  $c_e = w_e + \bar{\lambda}l_e$ .

Let  $\mathcal{S}$  be the set of spanning trees of minimum cost with respect to  $c_e^* = w_e + \bar{\lambda}^*l_e$ , where  $\bar{\lambda}^*$  is the value of  $\bar{\lambda}$  that maximizes  $l(\bar{\lambda})$ . For set  $\mathcal{S}$ , Ravi and Goemans [2] proved the following lemma.

**Lemma 2** *There is a spanning tree  $T \in \mathcal{S}$ , such that  $w(T)$  is at most  $LR$  and  $l(T) < L + l_{max}$ .*

Without loss of generality, assume that the length of each edge in  $G$  is not larger than  $L$ , i.e.,  $l_e \leq L$  for each edge  $e \in E$ . Thus,  $l_{max} \leq L$ . The reason is that any edge with length larger than  $L$  cannot be used to construct a spanning tree with total length not more than  $L$ . Therefore, according to Lemma 2, there is a spanning tree  $T \in \mathcal{S}$ , such that  $w(T)$  is at most  $LR$  and  $l(T) \leq L + l_{max}$ , which implies that  $w(T) \leq W$  and  $l(T) \leq 2L$  as  $LR \leq W$  and  $l_{max} \leq L$ , i.e.,  $T$  is a (2,1)-approximation solution. Since such a (2,1)-approximation solution is in  $\mathcal{S}$ , it corresponds to a line function  $g \in \bar{A}^*$ .

In the following, we will show that GSEMO can also find a (2,1)-approximation solution for the CMST problem in a pseudopolynomial runtime, under condition that the weight and length on each edge are positive integer numbers. Before that, we describe three useful lemmas and prove a lemma.

**Lemma 3** [38, 39] *Let  $T^*$  and  $T$  be the minimum spanning tree and an arbitrary spanning tree of a given connected graph  $G = (V, E)$ , respectively. Let  $E(T^*)$  and  $E(T)$  be the set of edges of  $T^*$  and  $T$ , respectively. Then, there exists a bijection  $\phi: E(T^*) \setminus E(T) \rightarrow E(T) \setminus E(T^*)$ , such that  $\phi(e) \in E(T) \setminus E(T^*)$  is on the cycle produced by including  $e \in E(T^*) \setminus E(T)$  into  $T$ , and  $w(\phi(e)) \geq w(e)$ .*

Lemma 3 tells us that an arbitrary spanning tree can be transformed into the minimum spanning tree by exchanging

at most  $n - 1$  pairs of edges between these two spanning trees, and each exchange cannot increase the weight of the spanning tree. Furthermore, we have the following Lemma.

**Lemma 4** [40] *Let  $x$  be a solution describing a spanning tree  $T$ . Then, there exists a set of  $n$  2-bit flips, such that the average weight decrease of these flips is at least  $(w(x) - w_{opt})/n$ .*

**Lemma 5** (Multiplicative Drift [41]) *Let  $S \subseteq \mathbb{R}$  be a finite set of positive numbers with minimum  $s_{min}$ . Let  $\{X^{(t)}\}_{t \in \mathbb{N}}$  be a sequence of random variables over  $S \cup \{0\}$ . Let  $\tau$  be the random variable that denotes the first point in time  $t \in \mathbb{N}$  for which  $X^{(t)} = 0$ .*

Suppose that there exists a constant  $\delta > 0$ , such that

$$E[X^{(t)} - X^{(t+1)} | X^{(t)} = s] > \delta s$$

holds for all  $s \in S$  with  $P[X^{(t)} = s] > 0$ . Then for all  $s_0 \in S$  with  $P[X^{(0)} = s_0] > 0$ ,

$$E[\tau | X^{(0)} = s_0] \leq \frac{1 + \ln(s_0/s_{min})}{\delta}.$$

Lemma 5 is usually used to estimate the runtime of evolutionary algorithms. Based on Lemmas 3 to 5, we can estimate the runtime for GSEMO to find a spanning tree for each objective vector on the convex hull of the Pareto front of the transformed BMST problem starting from any initial solution.

**Lemma 6** *The expected runtime for GSEMO starting from any initial solution to construct a population which includes a spanning tree for each objective vector on the convex hull of the Pareto front of the transformed BMST problem is  $O(nm^2 l_{wl}(\ln n + u_{wl}))$ .*

**Proof** Before a connected graph is found, the population contains only one solution, as the solution with the smallest number of connected components dominates that with a larger number of connected components. Let the current solution is  $x$  whose number of connected components is  $k$ , and then, its number of connected components can be reduced by one with probability at least  $\frac{k-1}{m}$ , which implies that the expected waiting time is at most  $\frac{m}{k-1}$ . Therefore, a connected graph will be included in the population in expected time  $O(m(\frac{1}{n-1} + \dots + \frac{1}{2} + 1)) = O(m \ln n)$ .

After a connected graph has been found, and before a spanning tree has been found, the population contains only one connected graph, as the connected graph with the smallest number of edges dominates that with a larger number of edges. Let the current connected graph is  $x$  whose number of edges is  $k$ , then its number of edges can be reduced by at

least one with probability at least  $\frac{k-(n-1)}{m}$ , which implies that the expected waiting time is  $O(\frac{m}{k-(n-1)})$ . Therefore, a spanning tree will be included in the population in expected time  $O(m(\frac{1}{m-(n-1)} + \dots + \frac{1}{2} + 1)) = O(m \ln(m-n+1)) = O(m \ln n)$  as  $m = O(n^2)$ .

Once the first one spanning tree has been included, the population accepts only spanning trees. Since for each value of  $\bar{f}_1$  and  $\bar{f}_2$ , there is only one spanning tree can be included, the population contains at most  $nl_{wl}$  spanning trees, i.e., population size is  $O(nl_{wl})$ , where  $l_{wl} = \min\{w_{max}, l_{max}\}$ .

Let  $x$  be the solution in population with the smallest value of  $\bar{f}_1$  ( $\bar{f}_2$ ), and  $w(x)$  be its weight, then it can be selected from population with probability  $\Omega(\frac{1}{nl_{wl}})$ , as the population size is  $O(nl_{wl})$ . Consider the progress  $\Delta^{(t)} = w(x^{(t)}) - w(x^{(t+1)}) = (w(x^{(t)}) - w_{opt}) - (w(x^{(t+1)}) - w_{opt})$  of the GSEMO in the  $t$ th generation. Furthermore, according to Lemma 4, the weight of  $x^{(t)}$  can be decreased by average of  $(w(x^{(t)}) - w_{opt})/n$  with probability  $\Theta(n/m^2)$ . Altogether, the probability that the weight of  $x^{(t)}$  can be decreased by average of  $(w(x^{(t)}) - w_{opt})/n$  is  $\Omega(\frac{1}{nl_{wl}} \cdot \frac{n}{m^2})$ . Thus,  $E[\Delta^{(t)} | x^{(t)} - w_{opt}] = \Omega((w(x^{(t)}) - w_{opt}) \cdot \frac{1}{nm^2 l_{wl}})$ . Then, according to Lemma 5, the minimum spanning tree with respect to  $\bar{f}_1$  ( $\bar{f}_2$ ) will be included in population by the GSEMO in expected time  $E[\tau] = O(nm^2 l_{wl}(1 + \ln(nw_{max})) (O(nm^2 l_{wl}(1 + \ln(nl_{max}))) = O(nm^2 l_{wl}(\ln n + \ln w_{max})) (O(nm^2 l_{wl}(\ln n + \ln l_{max})))$ .

After that, a spanning tree for  $p_0$  ( $p_h$ ) will be included in population in expected time  $O(nm^2 l_{wl}(\ln n + \ln w_{max})) (O(nm^2 l_{wl}(\ln n + \ln l_{max})))$  according to Lemma 5.

Now, we analyze the expected time for the GSEMO to construct a spanning tree for each objective vector on the convex hull of the Pareto front of the transformed BMST problem after a Pareto optimal solution  $p_0$  has been constructed.

Let  $T$  be a spanning tree corresponding to extremal point  $p_i$  ( $0 \leq i \leq h-1$ ), then an exchange of two edges may transform  $T$  into a spanning tree  $T'$  corresponding to extremal point  $p_{i+1}$  or a spanning tree on line  $l_i$  that connecting  $p_i$  and  $p_{i+1}$ . Note that the resulted spanning tree  $T'$  cannot locate

below line  $l_i$ . Suppose the resulted spanning tree  $T''$  be over line  $l_i$ , then the remaining exchanges that transform  $T''$  into  $T'$  can transform  $T$  into a spanning tree lying below line  $l_i$ , as  $T$  can be transformed into  $T'$  according to Lemmas 1 and 3. Suppose that there are  $h_i$  spanning trees on line  $l_i$ , then the GSEMO can transform  $T$  into  $T'$  in expected time  $O(m^2 nl_{wl}(h_i + 1))$ , as the probability of performing an exchange of two edges is  $\frac{1}{m^2}$  and the probability of selecting  $T$  from population is  $\frac{1}{nl_{wl}}$ .

Finally, the GSEMO constructs a spanning tree for each objective vector on the convex hull of the Pareto front of the transformed BMST problem in expected time  $O(m^2 nl_{wl} \sum_{i=0}^{h-1} (h_i + 1)) = O(m^2 nl_{wl} \cdot |H|)$ . Here,  $H$  is the set containing all objective vectors on the convex hull of the Pareto front of the transformed BMST problem, and  $|H|$  is the number of elements in  $H$ . Since both weight and length on each edge of  $G$  are positive integer numbers, and note that for each value of  $\bar{f}_1$  or  $\bar{f}_2$ , there exists a Pareto optimal solution, we have  $|H| \leq u_{wl}$ , where  $u_{wl} = \max\{w_{max}, l_{max}\}$ .

Altogether, the GSEMO constructs a spanning tree for each objective vector on the convex hull of the Pareto front of the transformed BMST problem in expected time  $O(nm^2 l_{wl}(\ln n + u_{wl}))$ . □

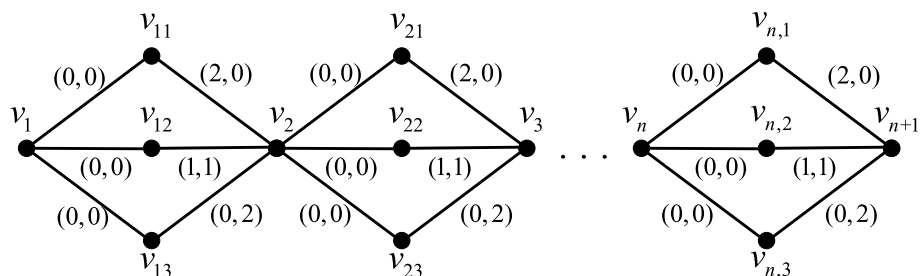
**Theorem 1** *GSEMO starting from any initial solution can construct a population which includes a (2,1)-approximation solution for the CMST problem in expected runtime  $O(nm^2 l_{wl}(\ln n + u_{wl}))$ .*

**Proof of Theorem 1** According to Lemma 2, there is a (2, 1)-approximation solution for the CMST problem in  $\mathcal{S}$ , which can be found by Lagrangian relaxation. Then, by Lemma 1, it can be found by the weighted-sums approach, which means that it is a solution with objective vector on the convex hull of the Pareto front of the transformed BMST problem.

According to Lemma 6, GSEMO can find a spanning tree for each objective vector on the convex hull of the transformed BMST problem in time  $O(nm^2 l_{wl}(\ln n + u_{wl}))$ .

Therefore, GSEMO can find a (2, 1)-approximation solution for the CMST problem in runtime  $O(nm^2 l_{wl}(\ln n + u_{wl}))$ . □

Fig. 3 An instance  $G'$  of the CMST problem



### 4 Instance that GSEMO Can Optimize Efficiently

In this section, an instance  $G' = (V', E')$  of the CMST problem is constructed to show that GSEMO can efficiently optimize some instance of the CMST problem. As shown in Fig. 3,  $V' = \{v_1, v_{11}, v_{12}, v_{13}, \dots, v_i, v_{i1}, v_{i2}, v_{i3}, \dots, v_n, v_{n,1}, v_{n,2}, v_{n,3}, v_{n+1}\}$ , and  $E'$  contains edges connecting  $v_i$  to  $v_{ij}$  and also edges connecting  $v_{ij}$  to  $v_{i+1}$  for  $i \in \{1, \dots, n\}$  and  $j \in \{1, 2, 3\}$ . Therefore, there are  $4n + 1$  nodes and  $6n$  edges in  $G'$ , i.e.,  $|V'| = 4n + 1$  and  $|E'| = 6n$ .

For an edge  $e \in E'$ , we call the vector  $(w_e, l_e)$  cost vector of  $e$ . The cost vector of edges  $(v_i, v_{ij})$  is  $(0, 0)$  for  $i \in \{1, \dots, n\}$  and  $j \in \{1, 2, 3\}$ . While the cost vectors of edges  $(v_{i1}, v_{i+1})$ ,  $(v_{i2}, v_{i+1})$ , and  $(v_{i3}, v_{i+1})$  are  $(2, 0)$ ,  $(1, 1)$ , and  $(0, 2)$ , respectively, for  $i \in \{1, \dots, n\}$ .

For this instance, we aim to find a spanning tree with minimum weight under the condition that its length is no more than  $n$ , that is

$$\begin{aligned} \min \quad & \sum_{e \in E(T)} w_e \\ \text{s.t.} \quad & \sum_{e \in E(T)} l_e \leq n, \\ & T \in \mathcal{T}. \end{aligned} \tag{5}$$

We first transform problem (5) into the following BMST problem, then use GSEMO to find all Pareto optimal solutions, and finally select one solution with minimal weight among those having lengths no more than  $n$

$$\begin{aligned} \min \quad & \sum_{e \in E(T)} w_e \\ \min \quad & \sum_{e \in E(T)} l_e \\ \text{s.t.} \quad & T \in \mathcal{T}. \end{aligned} \tag{6}$$

There are  $2n + 1$  Pareto optimal solutions whose objective vectors are  $(2n, 0), (2n - 1, 1), \dots, (1, 2n - 1), (0, 2n)$ , respectively. Each Pareto solution contains all edges of cost vector  $(0, 0)$ . Besides, a Pareto optimal solution with  $(2n - i, i)$  contains  $n - i$  edges of cost vector  $(2, 0)$  and  $i$  edges of cost vector  $(1, 1)$  if  $0 \leq i \leq n$ , or  $2n - i$  edges of cost vector  $(1, 1)$  and  $i - n$  edges of cost vector  $(0, 2)$  if  $n < i \leq 2n$ .

If GSEMO can find all Pareto optimal solutions of the problem (6), then we can select from the population a solution with minimum weight among all solutions with lengths no more than  $n$ . For  $G'$ , the optimal solution has cost vector  $(n, n)$ . In this sense, we say that GSEMO can find the optimal solution of the problem (5).

**Theorem 2** *When transforming the constrained MST problem to a BMST problem, GSEMO can find the optimal*

*solution for instance  $G'$  in expected runtime  $O(n^4)$  starting with any initial solution.*

**Proof** Note that  $|V'| = 4n + 1$  and  $|E'| = 6n$ . According to the proof of Lemma 6, starting with any initial solution GSEMO can construct a population consisting of connected subgraphs of  $G'$  in expected time  $O(n \ln n)$ . Moreover, according to the proof of Lemma 6, GSEMO can create a population consisting of spanning trees from a population of connected subgraphs in expected time  $O(n \ln n)$ .

If the population consists of spanning trees, then the population size is at most  $2n + 1$ , since in the population, there is at most one spanning tree for each objective vector  $(k, *)$ , where  $k$  is an integer and  $0 \leq k \leq 2n$ .

Now, we consider the spanning tree  $T$  whose weight  $w(T)$  is minimal among all spanning trees in the population. If  $w(T) > 0$ , then  $T$  contains at least one edge of cost vector  $(1, 1)$  or  $(2, 0)$ ; otherwise,  $w(T) = 0$  which contradicts that  $w(T) > 0$ .

Let edge  $(v_{i_0 j_0}, v_{i_0+1})$  be one of such edges with cost vector  $(2, 0)$  or  $(1, 1)$ , i.e.,  $i_0 \in \{1, \dots, n\}$  and  $j_0 = 1$  or  $2$ , contained in  $T$ . There are two cases needed to be considered with respect to whether edge  $(v_{i_0}, v_{i_0 j_0})$  is contained in  $T$  or not.

The first case is that edge  $(v_{i_0}, v_{i_0 j_0})$  is contained in  $T$ . In this case, one edge from  $\{(v_{i_0}, v_{i_0,3}), (v_{i_0,3}, v_{i_0+1})\}$  is not contained in  $T$  and the other is contained in  $T$ ; otherwise, there is a cycle consisting of  $v_{i_0}, v_{i_0 j_0}, v_{i_0,3}$  and  $v_{i_0+1}$  in  $T$  or node  $v_{i_0,3}$  is isolated. Adding the other edge to  $T$  and simultaneously removing edge  $(v_{i_0 j_0}, v_{i_0+1})$  will create a new spanning tree  $T'$  whose weight is at least one less than that of  $T$ .

The second case is that edge  $(v_{i_0}, v_{i_0 j_0})$  is not contained in  $T$ . In this case, adding edge  $(v_{i_0}, v_{i_0 j_0})$  and removing edge  $(v_{i_0 j_0}, v_{i_0+1})$  will produce a new spanning tree  $T'$  whose weight is at least one less than that of  $T$ .

No matter which case it is, a new spanning tree  $T'$  whose weight is at least one less than that of  $T$  can be created by adding one edge and simultaneously removing another one edge. Since there is no solution in the population dominates  $T'$  or  $f(T) = f(T')$ , it will be included in the population. The probability of choosing  $T$  to mutate is  $\Omega(\frac{1}{n})$  and the probability of producing  $T'$  from  $T$  is  $\Omega(\frac{1}{n^2})$ , which implies that  $T'$  can be included in the population in expected time  $O(n^3)$ . In other words, the weight of  $T$  can be decreased by at least one in expected time  $O(n^3)$ . Thus, a spanning tree with weight 0 will be created in expected time  $O(n^4)$  after a population consisting of spanning trees has been constructed, as the weight of any spanning tree is no more than  $2n$ , i.e.,  $0 \leq w(T) \leq 2n$ . The spanning tree with weight 0 is the Pareto optimal solution whose objective vector is  $(0, 2n)$ , since there is only one spanning tree with weight 0 and it consists of all edges with weight 0.



Assume that a Pareto optimal solution  $T$  with objective vector  $(i, 2n - i)$  ( $0 \leq i < n$ ) has been included in the population, then  $T$  consists of  $i$  edges of cost vector  $(1, 1)$  and  $n - i$  edges of cost vector  $(0, 2)$ , and all edges of cost vector  $(0,0)$ . A Pareto optimal solution  $T'$  with objective vector  $(i + 1, 2n - i - 1)$  can be created from  $T$  by adding one edge  $(v_{i,2}, v_{i+1})$  ( $i \in \{1, \dots, n\}$ ) with cost vector  $(1, 1)$  and removing the edge  $(v_{i,3}, v_{i+1})$  with cost vector  $(0, 2)$ . The probability of choosing  $T$  from the population is  $\Omega(\frac{1}{n})$  and the probability of creating  $T'$  from  $T$  is  $\Omega(\frac{1}{n^2})$ , so  $T'$  will be included in the population in expected time  $O(n^3)$  after  $T$  has been included. Therefore, Pareto optimal solutions with objective vectors  $(1, 2n - 1), \dots, (n, n)$  will be all included in the population in expected time  $O(n^4)$  after the Pareto optimal solution with objective vector  $(0, 2n)$  has been included.

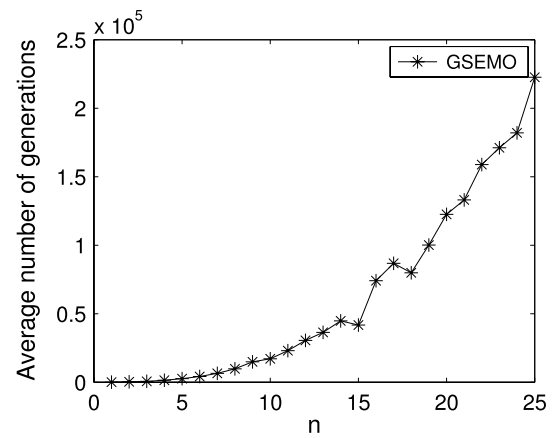
Assume that a Pareto optimal solution  $T$  with objective vector  $(n + i, n - i)$  ( $0 \leq i < n$ ) has been included in the population, then  $T$  consists of  $n - i$  edges of cost vector  $(1, 1)$  and  $i$  edges of cost vector  $(2, 0)$ , and all edges of cost vector  $(0,0)$ . A Pareto optimal solution  $T'$  with objective vector  $(n + i + 1, n - i - 1)$  can be created from  $T$  by adding one edge  $(v_{i,1}, v_{i+1})$  ( $i \in \{1, \dots, n\}$ ) with cost vector  $(2, 0)$  and removing the edge  $(v_{i,2}, v_{i+1})$ . Analogously,  $T'$  will be included in the population in expected time  $O(n^3)$  after  $T$  has been included. Furthermore, Pareto optimal solutions with objective vectors  $(n + 1, n - 1), \dots, (2n, 0)$  will be all included in the population in expected time  $O(n^4)$  after the Pareto optimal solution with objective vector  $(n, n)$  has been included.

Altogether, all Pareto optimal solutions with objective vectors  $(0, 2n), (1, 2n - 1), \dots, (n, n), \dots, (2n - 1, 1), (2n, 0)$  will be included in the population by GSEMO in expected runtime  $O(n^4)$  starting with any initial solution. □

We now experimentally verify the theoretical results on instance  $G'$ . Experimental computer is Intel(R) Core(TM)

**Table 1** Average number of generations (aveGen) over 30 independent runs for GSEMO to find the optimal solution to instance  $G'$  when transforming the CMST problem of instance  $G'$  to a BMST problem

$n$	1	2	3	4	5
aveGen	49	248	620	1461	2604
$n$	6	7	8	9	10
aveGen	4168	6466	9765	14907	17066
$n$	11	12	13	14	15
aveGen	23034	30549	36370	44763	41684
$n$	16	17	18	19	20
aveGen	74092	86775	79819	100132	122535
$n$	21	22	23	24	25
aveGen	133082	158832	171143	182017	222603



**Fig. 4** The curve of average number of generations over 30 independent runs for GSEMO to find the optimal solution to instance  $G'$  when  $n$  varies from 1 to 25, when the CMST problem of instance  $G'$  is transformed to a BMST problem

2.10-GHz with 2.0-GB RAM. Table 1 and Fig. 4 report the experimental results of 30 independent runs when  $n$  varies from 1 to 25, which show that by transforming the CMST problem of instance  $G'$  to a BMST problem, GSEMO can efficiently find the optimal solution when  $n$  varies from 1 to 25.

### 5 Conclusions and Discussions

Theoretical analysis of the performance of the EA on (NP-) hard problems is nowadays a hot topic. Especially, the approximation performance analysis of the EA on NP-hard combinatorial optimization problems recently attracts much attention from researchers.

Following this line of research, we analyze the approximation performance of GSEMO on the CMST problem. By analyzing the relationship between the CMST problem and its transformed BMST problem, we reveal that GSEMO can find a (2,1)-approximation solution for this problem in a pseudopolynomial runtime.

On a constructed instance of the CMST problem, we show that GSEMO is efficient to find its optimal solution. This illustrates that the MOEA is capable of finding the optimal solutions for some instances of the CMST problem.

The results of approximation performance analysis and the runtime analysis on the constructed instance show that GSEMO is suitable for the CMST problem.

In the future, we will investigate the performance of the MOEA on those CMST problems with two or more constraints, including the approximation performance.

**Acknowledgements** The authors would like to thank the anonymous reviewers for their valuable comments and suggestions that helped

improve the quality of this manuscript. We appreciate Prof. Yuren Zhou and Dr. Xiaoyun Xia for their constructive suggestions.

**Author Contributions** XL completed all the work of the paper.

**Funding** This work was supported in part by National Natural Science Foundation of China (Nos. 61562071, 61773410).

**Availability of Data and Materials** No applicable.

## Declarations

**Conflict of Interest** The authors declare that they have no competing interests.

**Ethics Approval and Consent to Participate** No applicable.

**Consent for Publication** No applicable.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Aggarwal, V., Aneja, Y.P., Nair, K.P.K.: Minimal spanning tree subject to a side constraint. *Comput. Oper. Res.* **9**(4), 287–296 (1982)
- Ravi, R., Goemans, M.X.: The constrained minimum spanning tree problem. Paper presented at the 5th Scandinavian workshop on algorithm theory, Reykjavik, Iceland, 3–5 July 1996 (1996)
- Hong, S.P., Chung, S.J., Park, B.H.: A fully polynomial bicriteria approximation scheme for the constrained spanning tree problem. *Oper. Res. Lett.* **32**(3), 233–239 (2004)
- Hassin, R., Levin, A.: An efficient polynomial time approximation scheme for the constrained minimum spanning tree problem using matroid intersection. *SIAM J. Comput.* **33**(2), 261–268 (2004)
- Chen, Y.H.: Polynomial time approximation schemes for the constrained minimum spanning tree problem. *J. Appl. Math.* **2012**(Article ID 394721) (2012). <https://doi.org/10.1155/2012/394721>
- Zou, N., Guo, L.: Improved approximating algorithms for computing energy constrained minimum cost steiner trees. Paper presented at the 15th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2015), 567–577 (2015)
- Michalewicz, Z., Schoenauer, M.: Evolutionary algorithm for constrained parameter optimization problems. *Evol. Comput.* **4**(1), 1–32 (1996)
- Michalewicz, Z., Deb, K., Schmidt, M., Stidsen, T.: Test-case generator for nonlinear continuous parameter optimization techniques. *IEEE Trans. Evol. Comput.* **4**(3), 197–215 (2000)
- Coello, C.A.C.: Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput. Methods Appl. Mech. Eng.* **191**(11–12), 1245–1287 (2002)
- Mezura-Montes, E., Coello, C.A.C.: A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Trans. Evol. Comput.* **9**(1), 1–17 (2005)
- Runarsson, T.P., X., Y.: Search biases in constrained evolutionary optimization. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **35**(2), 233–243 (2005)
- Wang, Y., Cai, Z.: A dynamic hybrid framework for constrained evolutionary optimization. *IEEE Trans. Syst. Man Cybern. Part B: Cybern.* **42**(1), 203–217 (2012)
- Melo, V.V.D., Iacca, G.: A modified covariance matrix adaptation evolution strategy with adaptive penalty function and restart for constrained optimization. *Expert Syst. Appl.* **41**(16), 7077–7094 (2014)
- Spettel, P., Beyer, H.G., Hellwig, M.: A covariance matrix self-adaptation evolution strategy for optimization under linear constraints. *IEEE Trans. Evol. Comput.* **23**(3), 514–524 (2019)
- Coello, C.A.C.: Treating constraints as objectives for single-objective evolutionary optimization. *Eng. Optim.* **32**(3), 275–308 (2000)
- Cai, Z., Wang, Y.: A multiobjective optimization-based evolutionary algorithm for constrained optimization. *IEEE Trans. Evol. Comput.* **10**(6), 658–675 (2006)
- Woldesenbet, Y.G., Yen, G.G., Tessema, B.G.: Constraint handling in multi-objective evolutionary optimization. *IEEE Trans. Evol. Comput.* **13**(3), 514–525 (2009)
- Wang, Y., Cai, Z.: Combining multiobjective optimization with differential evolution to solve constrained optimization problems. *IEEE Trans. Evol. Comput.* **16**(1), 117–134 (2012)
- Witt, C.: Worst-case and average-case approximations by simple randomized search heuristics. Paper presented at the 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, 24–26 February 2005 (2005)
- Oliveto, P.S., He, J., Yao, X.: Analysis of the (1+1)-EA for finding approximate solutions to vertex cover problems. *IEEE Trans. Evol. Comput.* **13**(5), 1006–1029 (2009)
- Friedrich, T., He, J., Hebbinghaus, N., Neumann, F., Witt, C.: Approximating covering problems by randomized search heuristics using multiobjective models. *Evol. Comput.* **18**(4), 617–633 (2010)
- Neumann, F., Reichel, J.: Approximating minimum multicuts by evolutionary multiobjective algorithms. Paper presented at the 10th International Conference on Parallel Problem Solving from Nature, Dortmund, Germany, 13–17 September 2008, 72–81 (2008)
- Yu, Y., Yao, X., Zhou, Z.H.: On the approximation ability of evolutionary optimization with application to minimum set cover. *Artif. Intell.* **180–181**, 20–33 (2012)
- Neumann, F.: Expected runtimes of a simple evolutionary algorithm for the multi-objective minimum spanning tree problem. *Eur. J. Oper. Res.* **181**(3), 1620–1629 (2007)
- Horoba, C.: Exploring the runtime of an evolutionary algorithm for the multi-objective shortest path problem. *Evol. Comput.* **18**(3), 357–381 (2010)
- Zhou, Y., He, J.: A runtime analysis of evolutionary algorithms for constrained optimization problems. *IEEE Trans. Evol. Comput.* **11**(5), 608–619 (2007)
- Qian, C., Yu, Y., Zhou, Z.H.: On constrained boolean pareto optimization. Paper presented at the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, 25–31 July 2015, 389–395 (2015)
- Liu, Z.-Z., Wang, Y.: Handling constrained multiobjective optimization problems with constraints in both the decision and objective spaces. *IEEE Trans. Evol. Comput.* **23**(5), 870–884 (2019)

29. Tian, Y., Zhang, T., Xiao, J., Zhang, X., Jin, Y.: A coevolutionary framework for constrained multiobjective optimization problems. *IEEE Trans. Evol. Comput.* **25**(1), 102–116 (2021)
30. Zha, Q.B., Dong, Y.C., Chiclana, F., Herrera-Viedma, E.: Consensus reaching in multiple attribute group decision making: a multi-stage optimization feedback mechanism with individual bounded confidences. *IEEE Trans. Fuzzy Syst.* (2021). <https://doi.org/10.1109/TFUZZ.2021.3113571>
31. Qian, C., Yu, Y., Zhou, Z.H.: An analysis on recombination in multi-objective evolutionary optimization. *Artif. Intell.* **204**, 99–119 (2013)
32. Levin, A., Woeginger, G.J.: The constrained minimum weighted sum of job completion times problem. *Math. Program.* **108**(1), 115–126 (2006)
33. Handler, G.Y., Zang, I.: A dual algorithm for the constrained shortest path problem. *Networks* **10**(4), 293–309 (1980)
34. Gass, S., Saaty, T.: The computational algorithm for the parametric objective function. *Nav. Res. Logist.* **2**(1–2), 39–45 (2010)
35. Knowles, J.D., Corne, D.W.: A comparison of encodings and algorithms for multiobjective minimum spanning tree problems. Paper presented at the IEEE Congress on Evolutionary Computation (CEC 2001), Seoul, South Korea, 27–30 May 2001 (2001)
36. Boyd, S., Vandenberghe, L.: *Convex Optimization*. Cambridge University Press, Cambridge (2004)
37. Klamroth, K., Tind, J.: Constrained optimization using multiple objective programming. *J. Global Optim.* **37**(3), 325–355 (2007)
38. Kano, M.: Maximum and kth maximal spanning trees of a weighted graph. *Combinatorica* **7**, 205–214 (1987)
39. Mayr, E.W., Plaxton, C.G.: On the spanning trees of weighted graphs. *Combinatorica* **12**, 433–447 (1992)
40. Neumann, F., Wegener, I.: Randomized local search, evolutionary algorithms, and the minimum spanning tree problem. *Theor. Comput. Sci.* **378**, 32–40 (2007)
41. Doerr, B., Johannsen, D., Winzen, C.: Multiplicative drift analysis. *Algorithmica* **64**(4), 673–697 (2012)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.