



HAL
open science

On the Kalai-Smorodinsky solutions for Bi-objective Spanning Tree Problem

Thanh Loan Nguyen, Viet Hung Nguyen, Minh Hieu Nguyen, Thi Viet Thanh Vu

► **To cite this version:**

Thanh Loan Nguyen, Viet Hung Nguyen, Minh Hieu Nguyen, Thi Viet Thanh Vu. On the Kalai-Smorodinsky solutions for Bi-objective Spanning Tree Problem. 2024. hal-04637851

HAL Id: hal-04637851

<https://hal.science/hal-04637851v1>

Preprint submitted on 7 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On the Kalai-Smorodinsky solutions for Bi-objective Spanning Tree Problem

Thanh Loan Nguyen¹, Viet Hung Nguyen¹, Minh Hieu Nguyen¹, and
Thi Viet Thanh Vu²

¹ INP Clermont Auvergne, Univ Clermont Auvergne, Mines Saint-Etienne, CNRS,
UMR 6158 LIMOS, 1 Rue de la Chebarde, Aubiere Cedex, France
{[thanh_loan.nguyen](mailto:thanh_loan.nguyen@uca.fr),[viet_hung.nguyen](mailto:viet_hung.nguyen@uca.fr),[minh_hieu.nguyen](mailto:minh_hieu.nguyen@uca.fr)}@uca.fr
² Hanoi Open University, Viet Nam
thanhvtv82@gmail.com

Abstract. This paper investigates the Bi-objective Spanning Tree Problem (BSTP), an extension of the fundamental Minimum Spanning Tree problem with diverse applications. In this problem, each edge of a given graph is associated with two distinct weights, and the objective is to find a spanning tree that simultaneously minimizes both total weights. As a specific case of bi-objective optimization, we may be interested in enumerating all the Pareto-optimal solutions of the BSTP. However, as the BSTP has been shown to be NP-hard, all the existing approaches to enumerate efficient solutions cannot achieve a polynomial CPU time. In this paper, we propose a novel approach for finding preferred efficient solutions where the worst of the ratios of the two objective values to their maximum possible values, respectively, is minimized. This approach is based on the *Kalai-Smorodinsky* (KS) solution, a well-known concept in cooperative game theory. More precisely, we consider an extension of the KS solution concept to the non-convex case that can be found by optimizing convex combinations of two objectives. We first characterize the properties of such KS solution(s) for the BSTP. Next, we present a weakly polynomial-time algorithm for finding the KS solution(s) for the BSTP. Finally, we showcase the computational results in some instances and discuss the results. Beyond the particular case of the BSTP, this paper offers an efficient and explainable approach for solving bi-objective combinatorial optimization problems.

Keywords: Minimum Spanning Tree Problem · Bi-Objective Combinatorial Optimization · Kalai-Smorodinsky solution

1 Introduction

The Minimum Spanning Tree (MST) problem is a classical problem in graph theory and combinatorial optimization. This problem has numerous applications in various fields, including network design and transportation. This work focuses on the Bi-objective Spanning Tree Problem (BSTP), an extension of the MST

problem where two objectives need to be optimized simultaneously. In this problem, each edge of a graph is associated with two weight values, and the goal is to find a spanning tree that efficiently minimizes both total weights. As a particular case of bi-objective optimization, for solving the BSTP, we may be interested in enumerating all the Pareto-optimal solutions (also called *efficient solutions*) of the BSTP (i.e., the solutions that are not worse than any other solution on both objectives). We recall a crucial concept for classifying efficient solutions, especially in non-convex optimization: the distinction between supported and unsupported efficient solutions. An efficient solution is supported if it is located on the boundary of the feasible set's convex hull, while an unsupported efficient solution is located in the interior of the feasible set's convex hull.

Since the feasible solution set of the BSTP is finite, several exact methods in the literature have been proposed to enumerate its efficient solutions. Particularly, Ramos et al. [14] introduced the multi-objective branch-and-cut method, Steiner and Radzik [17] proposed the two-phase method using the k -best algorithm, Sourd and Spanjaard [16] applied the branch-and-bound method, and more recently, Santos et al. [15] provided the labeling algorithm. Besides, several approximation algorithms were also proposed for the BSTP [8]. However, the BSTP has been shown to be NP-hard (see [2]), which means that all the existing approaches to enumerate the supported efficient solutions cannot achieve worst-case polynomial-time complexity. Hence, we focus on particular efficient solutions for which we have a good justification of the compromise between the two objectives, and especially, we have a (weakly) polynomial-time algorithm for finding them. In this paper, we consider the efficient solutions that achieve the minimum on the worst (or maximum) ratios of the two objective values to their maximum possible values when optimizing individually.

Researchers have extensively explored the application of game theory to bi-objective combinatorial optimization, showing its effectiveness in addressing competing objectives [1]. By framing the BSTP as a game, we can utilize game theory concepts like Nash equilibria, Pareto efficiency, Kalai-Smorodinsky solution, etc., to find solutions that consider trade-offs between the two objectives. Our approach for the BSTP is based on the Kalai-Smorodinsky solution, a well-known concept in cooperative game theory, which provides a method to resolve disputes or negotiations by finding a fair agreement between the players. The solution is particularly relevant in situations where they need to decide on the division of a scarce resource or any negotiation scenario where mutual agreements are sought [6]. The Kalai-Smorodinsky approach applied to the bi-objective optimization problem seeks efficient solutions that are balanced between the *Utopia point*, combining the (best-case) minimal values of the two objectives when minimized independently, and the *Nadir point*, representing the (worst-case) maximal values of the objectives. Notice that if the feasible set is convex, the solution given by the Kalai-Smorodinsky approach, called the *KS solution*, guarantees that each objective receives the same ratio of its minimum possible value over its maximum possible value [6]. Geometrically, it is a unique point located on the straight line connecting the Utopia and Nadir points (i.e., the Utopia-Nadir line)

in correspondence with the Pareto front, which contains all efficient solutions. Hence, the KS solution is the point lying on the Utopia-Nadir line and having a minimal distance from the Utopia point. Based on this point, some research in the literature has provided optimization algorithms for finding the KS solution [10, 5].

In contrast, since the Pareto front of the BSTP is generally non-convex, the Utopia-Nadir line may not intersect with the Pareto front. Thus, the KS solution described in the convex case may not exist for the BSTP. To address this issue, some approaches extended the KS solution concept to accommodate non-convex feasible sets (e.g., see [11, 18]). Although these studies have presented the concept and characterized the KS solution in such contexts, the question of how to algorithmically determine the KS solution for combinatorial optimization problems remains a challenge. In this paper, we focus on supported efficient solutions since they can provide important insights about the whole Pareto-optimal solution set. It should be noted that supported efficient solutions can be obtained by optimizing convex combinations of objectives [13]. We first present an extension of the KS solution concept in the context of the BSTP and then show that KS solution(s) exist in the supported efficient solution set of the BSTP. To identify these solutions, we develop a (weakly) polynomial-time algorithm that employs a binary search approach. This algorithm is designed to converge in a logarithmic number of iterations, where the logarithm is taken with respect to fixed parameters determined by the problem data. Lastly, we present practical computational results and provide insights to show the effectiveness of our proposed approach.

The remainder of this paper is organized as follows: Section 2 formally defines the BSTP and discusses the characterization of the KS solution(s) for it. Section 3 proposes a binary search algorithm for finding KS solution(s) in the supported solution set. Section 4 provides computational results on some instances of the BSTP and discusses them. Finally, Section 5 concludes the paper and outlines future work directions.

2 Kalai-Smorodinsky solution for BSTP

2.1 KS solution: A literature review

The KS solution is a concept from the field of game theory, particularly in the context of bargaining problems developed by Ehud Kalai and Meir Smorodinsky in 1975 [6]. This solution concept addresses how to fairly divide a set of resources or determine an outcome that multiple parties can agree upon. As an alternative to the well-known Nash bargaining solution [12], the key advantages of the KS solution are its emphasis on fairness and its applicability to a wide range of bargaining and optimization problems. In this paper, we focus on the KS solution for bi-objective optimization, particularly in the context of convex and non-convex feasible sets.

Let us consider a general bi-objective optimization problem which can be formulated as

$$\min_{x \in \mathcal{X}} (f_1(x), f_2(x)),$$

where \mathcal{X} denotes the set of all feasible decision vectors x . We also suppose that $f_1(x) > 0$ and $f_2(x) > 0, \forall x \in \mathcal{X}$. In general bi-objective optimization, the concepts of the Utopia and Nadir points are crucial for characterizing the KS solution. Let $\mathbf{x}_1^*, \mathbf{x}_2^* \in \mathcal{X}$ be the decision vectors which minimize f_1, f_2 , respectively (resp.). Let us consider the space \mathbb{R}_+^2 where the X -axis represents f_1 and the Y -axis represents f_2 . The Utopia point is the point where both objectives take minimum values, i.e., $(f_1(\mathbf{x}_1^*), f_2(\mathbf{x}_2^*))$. Then, the Nadir point is obtained by computing each objective with the optimal decision vector of the other objective, i.e., $(f_1(\mathbf{x}_2^*), f_2(\mathbf{x}_1^*))$. In order to effectively regularize the ratios of each objective's minimum over their maximum values, we first shift the two objectives f_1, f_2 with respect to the coordinates of the Nadir and Utopia points:

$$P(x) = f_1(x) - f_1(\mathbf{x}_1^*) \text{ and } Q(x) = f_2(x) - f_2(\mathbf{x}_2^*), \quad (1)$$

for all $x \in \mathcal{X}$. Let $(P, Q) = (P(x), Q(x))$ denote the objective values corresponding to a $x \in \mathcal{X}$. Let \mathcal{S} represent the set of pairs (P, Q) corresponding to all feasible decision vector solutions. This paper characterizes the feasible solutions for bi-objective combinatorial optimization using pairs (P, Q) instead of explicitly listing the decision vector solutions. Thus, two feasible solutions having the same values of (P, Q) will be considered equivalent. Throughout this paper, we use the notation " \equiv " to denote equivalent solutions. Furthermore, let $P_{max} = f_1(\mathbf{x}_2^*) - f_1(\mathbf{x}_1^*)$ and $Q_{max} = f_2(\mathbf{x}_1^*) - f_2(\mathbf{x}_2^*)$. Then, P_{max} (resp. Q_{max}) signifies the maximum value of objective P (resp. Q) for all efficient solutions (P, Q) . Given a solution (P, Q) , let us consider the ratios of $\rho_P = P/P_{max}$ and $\rho_Q = Q/Q_{max}$ which can be viewed as a measure of efficiency of P and Q , resp. Our objective is to find a Pareto-optimal solution that satisfies two criteria: fairness and efficiency. For fairness, we aim to minimize the fairness measure $|\rho_P - \rho_Q|$, bringing it as close to 0 as possible. For efficiency, we want the solution to be as close as possible to the Utopia point. Note that although the Utopia and Nadir points both have a fairness measure of 0, the Nadir point is not an efficient solution, and the Utopia point is not a feasible solution.

Motivated by this issue, we are particularly interested in the KS solution, which ensures both criteria of fairness and efficiency. We first recall the KS solution concept when the feasible solution set \mathcal{X} is a convex set. In this setting, the KS solution ensures each objective receives the same ratio of their minimum possible values over their maximum possible values [6]. This implies that the KS solution is a Pareto-optimal solution (P, Q) with the measure of fairness $|\rho_P - \rho_Q| = 0$. Geometrically, it is a unique solution located on the Utopia-Nadir line in correspondence with the Pareto front [5] (see Fig. 1a).

Definition 1 (KS solution for convex case [6]). $(P^*, Q^*) \in \mathcal{S}$ is the KS solution provided that

$$\rho_{P^*} = \rho_{Q^*} = \rho \quad \text{and,} \quad (2)$$

$$\rho \leq \max\{\rho_P, \rho_Q\} \quad \forall (P, Q) \in \mathcal{S}. \quad (3)$$

Notice that in Definition 1, condition (2) assures that the KS solution lies on the Utopia-Nadir line, while condition (3) brings the KS solution as close as possible to the Utopia point. In contrast, the Utopia-Nadir line may not intersect with the Pareto front in the combinatorial optimization problem. Thus, the KS solution described in the convex case may not exist in the non-convex case. Therefore, it is necessary to reform the concept to fit this setting. Naturally, we may be interested in the supported efficient solution that is close to the Utopia-Nadir line, meaning its measure of efficiency approaches 0.

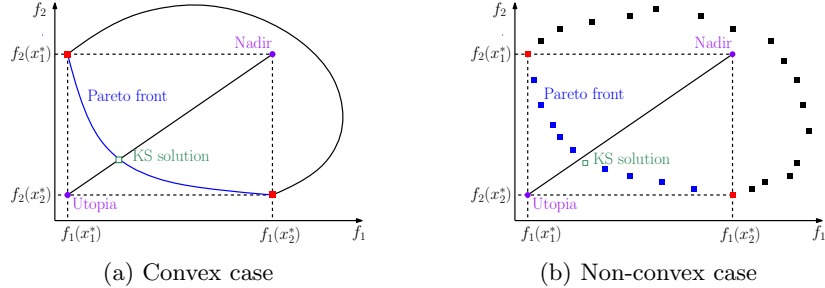


Fig. 1: Example of KS solution under convex and non-convex cases

We present the extension of the KS solution to the non-convex case according to the works of Nagahisa et al. [11] and Xu et al. [18]. In that setting, they proposed a relaxation for the KS solution concept without the condition (2). Then, when \mathcal{S} is non-convex, the KS solution(s) becomes the efficient solution(s) that achieves the minimum when we compare among all the feasible solutions the maximum of the two ratios of two objective values over their maximum possible values. We can formally state the definition of KS solution(s) as follows.

Definition 2 (KS solution for non-convex case [11, 18]). $(P^*, Q^*) \in \mathcal{S}$ is a KS solution provided that

$$\max\{\rho_{P^*}, \rho_{Q^*}\} \leq \max\{\rho_P, \rho_Q\}, \forall (P, Q) \in \mathcal{S}. \quad (4)$$

Intuitively, the KS solution(s) can be viewed as the best solution in terms of the ratio over the worst value for each objective. Fig. 1b illustrates an example of a KS solution where the feasible set is non-convex. In the next section, we will present the definition of the KS solution(s) for the BSTP and its properties.

2.2 KS solution for BSTP

This section begins by presenting the formal definition of the BSTP. Given an undirected graph $G = (V, E)$ where $|V| = n, |E| = m$ resp. is the set of nodes and edges. Let $c_e, t_e \in \mathbb{Z}_+ \setminus \{0\}$ denote two quantities which may be considered resp. as the cost and the time associated with each edge in G . The BSTP aims to find a spanning tree that minimizes both the total cost and total time, i.e.:

$$\min_{x \in \mathcal{T}} (f_1(x) = \sum_{e \in E} c_e x_e, \quad f_2(x) = \sum_{e \in E} t_e x_e),$$

where x_e is a binary variable that presents the occurrence of edge e in the spanning tree solution and the feasible solution set \mathcal{T} of G is the set of all vectors x that are associated with spanning trees in G .

Let $P(x), Q(x)$ be the shifted values of $f_1(x), f_2(x)$ resp. as defined in (1). Let \mathcal{S} be the set of all efficient solutions (P, Q) and $\mathcal{S}_{SE} \subseteq \mathcal{S}$ be the set of all supported efficient solutions of the BSTP. Since \mathcal{S} is finite and non-convex, \mathcal{S}_{SE} is also finite and non-convex. As a special case of the bi-objective combinatorial optimization, we precisely apply Definition 2 for the BSTP. Particularly, we state below the definition of the KS solution(s) for the BSTP over the supported efficient set \mathcal{S}_{SE} .

Definition 3 (KS solution for BSTP). $(P^*, Q^*) \in \mathcal{S}_{SE}$ is a KS solution provided that

$$\max\{\rho_{P^*}, \rho_{Q^*}\} \leq \max\{\rho_P, \rho_Q\}, \forall (P, Q) \in \mathcal{S}_{SE}. \quad (5)$$

Let $\tau = P_{max}/Q_{max} \in \mathbb{R}_+$ be the slope of the Utopia-Nadir line. Notice that the supported efficient set \mathcal{S}_{SE} can be divided into two subsets by the Utopia-Nadir line: $\mathcal{S}_{SE}^+ := \{(P, Q) \in \mathcal{S}_{SE} \mid P - \tau Q \geq 0\}$ and $\mathcal{S}_{SE}^- := \{(P, Q) \in \mathcal{S}_{SE} \mid P - \tau Q \leq 0\}$. According to the definitions provided, the proposition presented below demonstrates the existence of KS solution(s).

Proposition 1. *There always exists a KS solution for BSTP.*

Proof. Let $(P^+, Q^+) \in \mathcal{S}_{SE}^+$ and $(P^-, Q^-) \in \mathcal{S}_{SE}^-$ such that

$$(P^+, Q^+) = \underset{(P, Q) \in \mathcal{S}_{SE}^+}{\operatorname{argmin}} P - \tau Q, \quad (6a)$$

$$(P^-, Q^-) = \underset{(P, Q) \in \mathcal{S}_{SE}^-}{\operatorname{argmax}} P - \tau Q, \quad (6b)$$

According to (6a), $P^+ - \tau Q^+ \leq P - \tau Q, \forall (P, Q) \in \mathcal{S}_{SE}^+$. Since (P^+, Q^+) and (P, Q) are both Pareto-optimal solutions, we deduce $P^+ \leq P$ (otherwise, $P^+ > P, Q^+ < Q$ and then $P^+ - \tau Q^+ > P - \tau Q$ which leads to a contradiction). Thus, $\rho_{P^+} \leq \rho_P, \forall (P, Q) \in \mathcal{S}_{SE}^+$. Moreover, due to the definition of \mathcal{S}_{SE}^+ and $\tau = P_{max}/Q_{max}$, we get $\rho_{P^+} - \rho_{Q^+} = (P^+ - \tau Q^+)/P_{max} \geq 0, \forall (P, Q) \in \mathcal{S}_{SE}^+$. Similarly, we also have $\rho_{Q^-} \leq \rho_Q$ and $\rho_P \leq \rho_{Q^-}, \forall (P, Q) \in \mathcal{S}_{SE}^-$. Consequently,

$$\begin{aligned} \max\{\rho_{P^+}, \rho_{Q^+}\} &= \rho_{P^+} \leq \rho_P = \max\{\rho_P, \rho_Q\}, \forall (P, Q) \in \mathcal{S}_{SE}^+, \\ \max\{\rho_{P^-}, \rho_{Q^-}\} &= \rho_{Q^-} \leq \rho_Q = \max\{\rho_P, \rho_Q\}, \forall (P, Q) \in \mathcal{S}_{SE}^-. \end{aligned}$$

Hence, if $\rho_{P^+} < \rho_{Q^-}$ (resp. $\rho_{Q^-} < \rho_{P^+}$) then (P^+, Q^+) (resp. (P^-, Q^-)) is the unique KS solution. Otherwise, the KS solution set consists of both $(P^+, Q^+), (P^-, Q^-)$. That concludes the proof. \square

Notice that if $(P^+, Q^+) \equiv (P^-, Q^-)$, then there is a unique KS solution as the intersection between the Pareto front and the Utopia-Nadir line which coincides the convex case. If $(P^+, Q^+) \not\equiv (P^-, Q^-)$ then they are two consecutive supported efficient solutions that are located on two sides of the Utopia-Nadir line. Based on the proof of Proposition 1, the KS solution(s) must be selected from these two solutions, and there might be (at most) two distinct KS solutions. In the rest of this paper, we denote \mathcal{A} the set containing the solutions (P^+, Q^+) and (P^-, Q^-) . In the next section, we design a binary search algorithm for determining the KS solution(s) for the BSTP.

3 Algorithm for finding KS solution(s)

3.1 Algorithm construction

Let us state the sketch of our algorithm in this section. We recall that each supported efficient solution is necessarily a solution of a weighted sum single-objective optimization problem [13]

$$\mathcal{F}(\alpha) = \min (1 - \alpha)P + \alpha Q, \quad \forall (P, Q) \in \mathcal{S}, \quad \text{where } \alpha \in [0, 1].$$

Our algorithm is based on a binary search algorithm in the interval $[0, 1]$. Given an interval $[\alpha_i, \alpha_j] \subseteq [0, 1]$ and two solutions $(P_i, Q_i) \in \mathcal{S}_{SE}^-, (P_j, Q_j) \in \mathcal{S}_{SE}^+$ of resp. $\mathcal{F}(\alpha_i), \mathcal{F}(\alpha_j)$. We present Procedure SEARCH() to find the KS solution(s) which are solution(s) of $\mathcal{F}(\alpha)$ with $\alpha \in [\alpha_i, \alpha_j]$. We first calculate the midpoint α_m of this interval and then solve $\mathcal{F}(\alpha_m)$ to obtain a solution (P_m, Q_m) and verify that (P_m, Q_m) belongs to either the set \mathcal{S}_{SE}^+ or \mathcal{S}_{SE}^- . Then, we retain only one half-interval for further exploration in Procedure SEARCH(), let us say $[\alpha_i, \alpha_m]$ such that the two solutions (P_m, Q_m) and (P_i, Q_i) lie on two sides of the Utopia-Nadir line. This ensures that after each iteration, the two endpoints of the obtained interval correspond to two supported solutions situated on opposite sides of the Utopia-Nadir line. We continue these steps until we obtain an interval with a length smaller than a positive parameter ϵ defined by the input of the BSTP. The choice of ϵ guarantees the determination of two consecutive supported efficient solutions lying on two sides of the Utopia-Nadir line, i.e., two solutions in set \mathcal{A} . Our algorithm terminates when we obtain KS solution(s) through these two solutions. Consequently, it converges in a logarithmic number of iterations with respect to ϵ .

3.2 Algorithm statement

This section first states Lemma 1 to show the monotonic relationship between the value $\alpha \in [0, 1]$ and a solution (P, Q) of $\mathcal{F}(\alpha)$.

Lemma 1. *Given $0 \leq \alpha_i < \alpha_j \leq 1$. Let (P_i, Q_i) and (P_j, Q_j) be the solutions of resp. $\mathcal{F}(\alpha_i)$ and $\mathcal{F}(\alpha_j)$. Then, $P_i \leq P_j$ and $Q_i \geq Q_j$.*

Proof. Since (P_i, Q_i) and (P_j, Q_j) are the solutions of $\mathcal{F}(\alpha_i)$ and $\mathcal{F}(\alpha_j)$, resp., we deduce:

$$(1 - \alpha_i)P_i + \alpha_i Q_i \leq (1 - \alpha_i)P_j + \alpha_i Q_j, \text{ and} \quad (8a)$$

$$(1 - \alpha_j)P_j + \alpha_j Q_j \leq (1 - \alpha_j)P_i + \alpha_j Q_i \quad (8b)$$

Multiplying both sides of (8a) by $1 - \alpha_j \geq 0$ and (8b) by $1 - \alpha_i \geq 0$ then adding them, we obtain $(\alpha_i - \alpha_j)(Q_i - Q_j) \leq 0$. As $\alpha_i < \alpha_j$, this implies $Q_i \geq Q_j$. Moreover, from (8a), we have $(1 - \alpha_i)(P_i - P_j) \leq \alpha_i(Q_j - Q_i) \leq 0$. As $\alpha_i < 1$, we deduce $P_i \leq P_j$. \square

Due to Lemma 1, if $\alpha_i < \alpha_j$ and (P_i, Q_i) is the solution for both $\mathcal{F}(\alpha_i)$ and $\mathcal{F}(\alpha_j)$, then (P_i, Q_i) remains the solution for $\mathcal{F}(\alpha)$ for all $\alpha_i < \alpha < \alpha_j$. Building on this point, we present a method to determine elements in the set \mathcal{A} that are found by solving $\mathcal{F}(\alpha_i)$ and $\mathcal{F}(\alpha_j)$. A key is the determination of the length of $[\alpha_i, \alpha_j]$, which plays a role in the stopping condition for the iterative process.

Theorem 1. *Let $(P_i, Q_i) \in \mathcal{S}_{SE}^-$, $(P_j, Q_j) \in \mathcal{S}_{SE}^+$ be resp. the solutions of $\mathcal{F}(\alpha_i)$, $\mathcal{F}(\alpha_j)$. Suppose that $(P_i, Q_i) \not\equiv (P_j, Q_j)$ and $0 < \alpha_j - \alpha_i < \epsilon = 4/(P_{max} + Q_{max})^2$. Let $\alpha_s = \frac{P_j - P_i}{P_j - P_i + Q_i - Q_j}$ and (P_s, Q_s) be the solution of $\mathcal{F}(\alpha_s)$. Then,*

1. *If $(P_s, Q_s) \equiv (P_i, Q_i)$ or $(P_s, Q_s) \equiv (P_j, Q_j)$, $\mathcal{A} = \{(P_i, Q_i), (P_j, Q_j)\}$;*
2. *Otherwise, $\mathcal{A} = \{(P_i, Q_i), (P_s, Q_s)\}$ or $\mathcal{A} = \{(P_s, Q_s), (P_j, Q_j)\}$.*

Proof. We first show that α_s is well defined. Since $\alpha_i < \alpha_j$ and $(P_i, Q_i) \not\equiv (P_j, Q_j)$, $P_i < P_j$ and $Q_i > Q_j$ due to Lemma 1. Thus, $P_j - P_i + Q_i - Q_j > 0$.

We now show that $\alpha_s \in [\alpha_i, \alpha_j]$. Indeed, the optimalities of (P_i, Q_i) and (P_j, Q_j) give

$$(1 - \alpha_i)P_i + \alpha_i Q_i \leq (1 - \alpha_i)P_j + \alpha_i Q_j, \quad (9a)$$

$$(1 - \alpha_j)P_j + \alpha_j Q_j \leq (1 - \alpha_j)P_i + \alpha_j Q_i, \quad (9b)$$

From (9a) and (9b), we get

$$\alpha_i \leq \frac{P_j - P_i}{P_j - P_i + Q_i - Q_j} \leq \alpha_j,$$

which implies $\alpha_i \leq \alpha_s \leq \alpha_j$. We consider two following cases.

1. $(P_s, Q_s) \equiv (P_i, Q_i)$ or $(P_s, Q_s) \equiv (P_j, Q_j)$
 Without loss of generality, we assume that $(P_s, Q_s) \equiv (P_i, Q_i)$. Thus, (P_i, Q_i) is a solution of $\mathcal{F}(\alpha_s)$. The definition of α_s yields $(1 - \alpha_s)P_j + \alpha_s Q_j = (1 - \alpha_s)P_i + \alpha_s Q_i$. Therefore, (P_j, Q_j) is also a solution of $\mathcal{F}(\alpha_s)$. As a result of Lemma 1, (P_i, Q_i) (resp. (P_j, Q_j)) is the solution of $\mathcal{F}(\alpha)$ for all $\alpha \in (\alpha_i, \alpha_s]$ (resp. $\alpha \in (\alpha_s, \alpha_j)$). Thus, $\mathcal{A} = \{(P_i, Q_i), (P_j, Q_j)\}$.

2. $(P_s, Q_s) \neq (P_i, Q_i)$ and $(P_s, Q_s) \neq (P_j, Q_j)$

By the same argument, we also get $P_i < P_s < P_j, Q_i > Q_s > Q_j$ and

$$\alpha_i \leq \frac{P_s - P_i}{P_s - P_i + Q_i - Q_s} \leq \alpha_s \leq \frac{P_j - P_s}{P_j - P_s + Q_s - Q_j} \leq \alpha_j.$$

Suppose that $\frac{P_j - P_s}{P_j - P_s + Q_s - Q_j} > \frac{P_s - P_i}{P_s - P_i + Q_i - Q_s}$ then

$$(P_j - P_s)(P_s - P_i + Q_i - Q_s) > (P_s - P_i)(P_j - P_s + Q_s - Q_j)$$

which leads to

$$(P_j - P_s)(P_s - P_i + Q_i - Q_s) - (P_s - P_i)(P_j - P_s + Q_s - Q_j) \geq 1,$$

due to $P_i, P_j, P_s, Q_i, Q_j, Q_s \in \mathbb{Z}_+$.

Furthermore, the Cauchy-Schwarz inequality gives

$$\begin{aligned} (P_j - P_s + Q_s - Q_j)(P_s - P_i + Q_i - Q_s) &\leq \frac{(P_j + Q_i - P_i - Q_j)^2}{4} \\ &\leq \frac{(P_{\max} + Q_{\max})^2}{4}. \end{aligned}$$

Thus, we obtain

$$\begin{aligned} \alpha_j - \alpha_i &\geq \frac{P_j - P_s}{P_j - P_s + Q_s - Q_j} - \frac{P_s - P_i}{P_s - P_i + Q_i - Q_s} \\ &= \frac{(P_j - P_s)(P_s - P_i + Q_i - Q_s) - (P_s - P_i)(P_j - P_s + Q_s - Q_j)}{(P_j - P_s + Q_s - Q_j)(P_s - P_i + Q_i - Q_s)} \\ &\geq \frac{4}{(P_{\max} + Q_{\max})^2} = \epsilon. \end{aligned}$$

This leads to a contradiction. Hence,

$$\begin{aligned} \frac{P_j - P_s}{P_j - P_s + Q_s - Q_j} = \alpha_s = \frac{P_s - P_i}{P_s - P_i + Q_i - Q_s} \\ \Leftrightarrow (1 - \alpha_s)P_s + \alpha_s Q_s = (1 - \alpha_s)P_j + \alpha_s Q_j = (1 - \alpha_s)P_i + \alpha_s Q_i. \end{aligned}$$

This deduce that (P_i, Q_i) and (P_j, Q_j) are two solutions of $\mathcal{F}(\alpha_s)$. Similarly to the previous case, (P_i, Q_i) (resp. (P_j, Q_j)) is the solution of $\mathcal{F}(\alpha)$ for all $\alpha \in (\alpha_i, \alpha_s)$ (resp. $\alpha \in (\alpha_s, \alpha_j)$). Thus, $\mathcal{A} = \{(P_i, Q_i), (P_s, Q_s)\}$ or $\mathcal{A} = \{(P_s, Q_s), (P_j, Q_j)\}$. \square

Because of Lemma 1, we observe that a solution (P, Q) of $\mathcal{F}(\alpha)$ for all $\alpha \in [0, \alpha_i]$ (resp. $\alpha \in [\alpha_j, 1]$) belongs to \mathcal{S}_{SE}^- (resp. \mathcal{S}_{SE}^+), where α_i, α_j as defined in Theorem 1. Therefore, as a result of Theorem 1, we conclude that the interval $[0, 1]$ can be partitioned into two half-intervals: one associated with solutions in \mathcal{S}_{SE}^- and the other associated with solutions in \mathcal{S}_{SE}^+ . Let $c_{\max} = \max_{e \in E} c_e$ and

$t_{\max} = \max_{e \in E} t_e$. Since every spanning tree of the graph G has exactly $n - 1$ edges, we then obtain $P_{\max} \leq \max f_1(x) \leq (n - 1)c_{\max}$ and $Q_{\max} \leq \max f_2(x) \leq (n - 1)t_{\max}$. Consequently, we can take $\epsilon = 4/((n - 1)(c_{\max} + t_{\max}))^2$.

Given an interval $[\alpha_i, \alpha_j]$, we propose Procedure SEARCH() to find spanning tree(s) associated with KS solution(s) obtained by solving $\mathcal{F}(\alpha)$ with $\alpha \in [\alpha_i, \alpha_j]$.

Procedure 1 Find spanning tree(s) associated with KS solution(s) obtained by solving $\mathcal{F}(\alpha)$ where $\alpha \in [\alpha_i, \alpha_j]$

Input: • Two coefficients $0 \leq \alpha_i < \alpha_j \leq 1$ and two parameters τ, ϵ as defined above.

- 1: • $(P_i, Q_i) \in \mathcal{S}_{SE}^-$ corresponding with tree T_i is a solution of $\mathcal{F}(\alpha_i)$.
- 2: • $(P_j, Q_j) \in \mathcal{S}_{SE}^+$ corresponding with tree T_j is a solution of $\mathcal{F}(\alpha_j)$.
- 3: • $(P_i, Q_i) \neq (P_j, Q_j)$.

Output: Spanning tree(s) associated with the KS solution(s).

- 4: **procedure** SEARCH($\alpha_i, \alpha_j, \epsilon, \tau$)
- 5: **if** $\alpha_j - \alpha_i \geq \epsilon$ **then**
- 6: $\alpha_m \leftarrow (\alpha_i + \alpha_j)/2$
- 7: solving $\mathcal{F}(\alpha_m)$ to obtain (P_m, Q_m) and a corresponding spanning tree T_m
- 8: **if** $P_m - \tau Q_m == 0$ **then** return (P_m, Q_m) and T_m
- 9: **else if** $(P_m, Q_m) \in \mathcal{S}_{SE}^+$ **then** return SEARCH($\alpha_i, \alpha_m, \epsilon, \tau$)
- 10: **else** return SEARCH($\alpha_m, \alpha_j, \epsilon, \tau$)
- 11: **end if**
- 12: **else**
- 13: $\alpha_s \leftarrow (P_j - P_i)/(P_j - P_i + Q_i - Q_j)$
- 14: solving $\mathcal{F}(\alpha_s)$ to obtain (P_s, Q_s) and a corresponding spanning tree T_s
- 15: **if** $(P_s, Q_s) \in \mathcal{S}_{SE}^+$ **then**
- 16: **if** $\tau Q_i < P_s$ **then** return (P_i, Q_i) and T_i
- 17: **else if** $\tau Q_i > P_s$ **then** return (P_s, Q_s) and T_s
- 18: **else** return $(P_i, Q_i), (P_s, Q_s)$ and T_i, T_s
- 19: **end if**
- 20: **else**
- 21: **if** $\tau Q_s < P_j$ **then** return (P_s, Q_s) and T_s
- 22: **else if** $\tau Q_s > P_j$ **then** return (P_j, Q_j) and T_j
- 23: **else** return $(P_s, Q_s), (P_j, Q_j)$ and T_s, T_j
- 24: **end if**
- 25: **end if**
- 26: **end if**
- 27: **end procedure**

Based on this Procedure, we present the formal algorithm statement for determining the spanning tree(s) associated with the KS solution(s) for the BSTP and convergence proofs.

Algorithm 1 Find spanning tree(s) associated with KS solution(s) for BSTP

Input: A graph $G = (V, E)$ with $|V| = n$ and two edge weight vectors $c, t \in \mathbb{Z}_+^{|E|}$.
Output: Spanning tree(s) associated with KS solution(s) for the BSTP on G .
1: solving $\mathcal{F}(0), \mathcal{F}(1)$ to obtain $(P_0, Q_0), (P_1, Q_1)$, resp., and corresponding spanning trees T_0, T_1
2: **if** $(P_0, Q_0) \equiv (P_1, Q_1)$ **then** return (P_0, Q_0) and T_0
3: **else**
4: $c_{\max} \leftarrow \max_{e \in E} c_e; t_{\max} \leftarrow \max_{e \in E} t_e$
5: $\epsilon \leftarrow 4/((n-1)(c_{\max} + t_{\max}))^2; \tau \leftarrow P_1/Q_0$
6: return $SEARCH(0, 1, \epsilon, \tau)$
7: **end if**

Remark that due to the definition of P, Q by shifting, the solution $(P_0, Q_0) = (0, Q_0)$ belongs to \mathcal{S}_{SE}^- and $(P_1, Q_1) = (P_1, 0)$ belongs to \mathcal{S}_{SE}^+ . Moreover, the scalarization process only affects the values of the objective functions, while the optimal spanning tree associated with the KS solution(s) remains unchanged.

Theorem 2. *Algorithm 1 finds the KS solution(s) for the BSTP in weakly polynomial-time.*

Proof. We observe that Algorithm 1 performs a binary search over the interval $[0, 1]$ to locate the KS solution(s), which are the solution(s) of $\mathcal{F}(\alpha)$ with the coefficient $\alpha \in [0, 1]$. Specifically, at each iteration, the current search interval is split into two equal-length subintervals. The half-interval that is guaranteed to not contain α is discarded, and the search continues over the remaining half. The algorithm terminates when either the KS solution(s) is obtained or the length of the remaining search interval falls below ϵ . Since the interval size reduces by half at each iteration, the number of iterations does not exceed $\lceil \log(1/\epsilon) \rceil = \lceil 2 \log((n-1)(c_{\max} + t_{\max})) - 2 \rceil$, where the logarithm is taken with base 2.

We now show that $\mathcal{F}(\alpha)$ can be solved in polynomial-time where $\alpha \in [0, 1]$ is given. Due to the scalarization of P and Q , for each value α , minimizing $\mathcal{F}(\alpha) = (1-\alpha)P + \alpha Q$ is equivalent to minimizing $(1-\alpha)f_1(x) + \alpha f_2(x) = \sum_{e \in E} ((1-\alpha)c_e + \alpha t_e)x_e$, where $x \in \mathcal{T}$. Hence, we construct a graph $G' = (V, E)$ with the same sets of nodes and edges in G , and $(1-\alpha)c_e + \alpha t_e$ is a weight associated with each edge e of G' . Thus, solving $\mathcal{F}(\alpha)$ is now equivalent to solving the classical single-objective MST problem in G' .

In conclusion, Algorithm 1 determines the KS solution(s) in weakly polynomial-time, with a runtime of $\mathcal{C} \lceil 2 \log((n-1)(c_{\max} + t_{\max})) - 2 \rceil$, where \mathcal{C} represents the time complexity of the MST problem. \square

4 Computational results

We evaluated the performance of our proposed algorithm for the BSTP using randomly generated graphs from the NetworkX library. Specifically, we utilize the $G_{n,p}$ model to create Erdős-Rényi graphs, also known as binomial graphs [3].

In this model, n represents the number of nodes, and p denotes the probability of edge creation. For our experiments, we vary the number of nodes from 50 to 600 with a step size of 50. For each value of n , we create a graph “Ins n ” with an edge creation probability of $p = 0.4$. The edge cost values and time values were randomly from a uniform distribution over the interval resp. $[20, 30]$ and $[1, 10]$.

Instance	Min f_1			Min f_2			KS solution(s) for BSTP			
	f_1	f_2	Time (s)	f_1	f_2	Time (s)	f_1	f_2	Time (s)	Iterations
Ins50	986	276	0.01	1177	59	0.01	1024	105	0.31	22
Ins100	1981	498	0.01	2439	99	0.01	2054	159	1.32	24
Ins150	2980	767	0.01	3639	149	0.01	3038	204	2.87	25
Ins200	3980	1006	0.02	4828	199	0.02	4051	266	5.97	26
Ins250	4980	1156	0.02	6044	249	0.05	5041	302	10.24	27
Ins300	5980	1548	0.04	7292	299	0.06	6031	343	13.98	27
Ins350	6980	1683	0.05	8576	349	0.05	7017	401	19.66	28
Ins400	7980	2034	0.06	9777	399	0.06	8032	447	26.44	28
Ins450	8980	2182	0.08	11070	449	0.08	9018	487	33.97	29
Ins500	9980	2500	0.12	12163	499	0.12	10003	532	42.76	29
Ins550	10980	2777	0.15	13530	549	0.14	11006	573	49.97	29
Ins600	11980	2871	0.18	14720	599	0.14	12007	621	58.43	29

Table 1: Computational results for BSTP

Let us denote Min f_1 (resp. Min f_2) as the MST problem for the objective f_1 (resp. f_2). To solve the single-objective MST problem, we utilized Kruskal’s algorithm as implemented in the NetworkX package version 2.5.1 [4]. Table 1 presents the optimal solutions for the Min f_1 , Min f_2 , as well as the KS solution(s) for the BSTP. Notice that the solutions (f_1, f_2) for each problem Min f_1 and Min f_2 are to determine the domain for feasible solutions of the BSTP and to identify the Utopia and Nadir points. Also, the computation time and the number of iterations required to obtain each solution are shown in the columns “Time” and “Iterations”. Finally, Fig. 2 illustrates the KS solution for Ins150 and the set of supported solutions obtained using the method proposed by Steiner and Radzik [17]. The experiments were conducted on a system equipped with an Intel Core i5-10500 CPU, 3.10 GHz 15GB RAM with 6 cores and 12 threads.

The KS solution(s) offer a fair ratio between two objectives f_1 and f_2 over their maximum possible values than the optimal solution of the problem minimizing only f_1 or f_2 , as shown in Table 1. An interesting observation is that although there are theoretically at most two KS solutions for the BSTP, we obtained a unique KS solution for each instance in Table 1. The table also shows that our algorithm converges quickly following the time. The results in Table 1 demonstrate that our algorithm’s computational efficiency is remarkable, as it successfully solved instances with up to 600 vertices in a minute.

Fig. 2 shows that the KS solution is close to the Utopia-Nadir line, effectively balancing the two objectives, f_1 and f_2 . Furthermore, although the number of

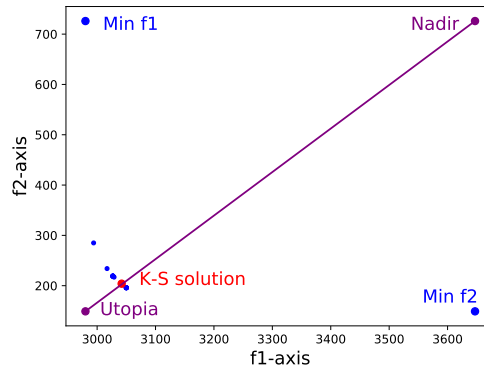


Fig. 2: The supported solutions with the KS solution in red.

supported solutions for the BSTP may be exponential, our algorithm explored only a subset of these solutions, shown as blue points in Fig. 2, thereby reducing the computational complexity. One challenge in solving the BSTP is the large number of iterations required. Further analysis shows that the supported solutions obtained in some iterations are equivalent, leading to redundant computations. This is clearly illustrated in Fig. 2, where the number of iterations is 25, but the number of distinct supported solutions obtained is only 7. Hence, the CPU time required to solve the BSTP increases significantly. Consequently, an interesting direction for future work would be to improve the stopping condition and refine the algorithm's convergence speed.

5 Conclusion

This paper explored the Bi-objective Spanning Tree Problem (BSTP), aiming to find a spanning tree that minimizes both the total cost and the total time. Then, we proposed a novel approach for solving the BSTP, utilizing the Kalai-Smorodinsky (KS) solution concept from cooperative game theory. The main contributions of this paper include the characterization of KS solution(s) for the BSTP and the development of an exact algorithm that determines KS solution(s) in polynomial-time. Finally, computational experiments in some instances have shown the effectiveness of our algorithm, indicating its rapid convergence.

A significant remark is that this approach can be applied to a general bi-objective combinatorial optimization problem. Hence, it can be a standard criterion for solving bi-objective combinatorial optimization, where each objective can be maximized or minimized. Moreover, we aim to design an efficient algorithm for finding KS solution(s) in the Pareto front which contains both supported and non-supported efficient solutions.

References

1. I.J. Curiel. Cooperative Game Theory and Applications: Cooperative Games Arising from Combinatorial Optimization Problems. 1997.
2. M. Ehrgott. Approximation algorithms for combinatorial multicriteria optimization problems. *Int. Trans. Oper. Res.* **7** 5–31, 2000.
3. P. Erdos and A. Renyi. On random graphs. *PUBL MATH-DEBRECEN*. **6** 290-297, 1959.
4. A.A. Hagberg, D.A. Schult and P.J. Swart. Exploring network structure, dynamics, and function using networkx. *Proc. Python Sci. Conf.*. 11–15, 2008.
5. L. Iorio, L. Fourment, S. Marie and M. Strano. Application of the Kalai-Smorodinsky approach in multi-objective optimization of metal forming processes. *Int. J. Mater. Form.*. **10** 515-526, 2017.
6. E. Kalai and M. Smorodinsky. Other Solutions to Nash’s Bargaining Problem. *Econometrica*. **43** (3) 513-518, 1975.
7. O. Kariv and L. Hakimi. An algorithmic approach to network location problems. ii: the p-medians. *Operations Research*, **37** (3) 539–560, 1979.
8. J.D. Knowles and D.W. Corne. A comparison of encodings and algorithms for multi-objective spanning tree problems. *Proc. IEEE CEC*. 544–55, 2001.
9. R. Kohli and H. Park. A cooperative game theory model of quantity discounts. *Manag. Sci.*. **35** (6) 693-707, 1989.
10. H. Moulin. Implementing the Kalai-Smorodinsky bargaining solution. *J. Econ. Theory*. **33** (1) 32-45, 1984.
11. R. Nagahisa and M. Tanaka. An axiomatization of the Kalai-Smorodinsky solution when the feasible sets can be finite. *Social Choice and Welfare*. **19** 751-761, 2002.
12. Nash, J.F. The bargaining problem. *Econometrica*. **18**, Issue 2, April 1950.
13. O. Özpeynirci and M. Köksalan. An exact algorithm for finding extreme supported non-dominated points of multiobjective mixed integer problems. *Manag. Sci.*. **56** (12) 2302–2315, 2010.
14. R. M. Ramos, S. Alonso, J. Sicilia and C. Gonzales. The problem of the optimal biobjective spanning tree. *Eur. J. Oper. Res.*. **111** 617–628, 1998.
15. J.L. Santos, L.D.P. Pugliese and F. Guerriero. A new approach for the multiobjective mini-mum spanning tree. *Comput. Oper. Res.*. (98) 69–83, 2018.
16. F. Sourd and O. Spanjaard. A Multiobjective Branch-and-Bound Framework: Application to the Biobjective Spanning Tree Problem. *INFORMS J. Comput.*. **20** (3), 472-484, 2008.
17. S. Steiner and T. Radzik. Computing all efficient solutions of the biobjective minimum spanning tree problem. *Comput. Oper. Res.*. **35** 198–211, 2008.
18. Y. Xu and N. Yoshihara. An equitable Nash solution to nonconvex bargaining problems. *Int. J. Game Theory*. **48** 769-779, 2019.