A Kalai-Smorodinsky approach for bi-objective combinatorial optimization

Thanh Loan Nguyen[1], Viet Hung Nguyen[1*], Minh Hieu Nguyen[2]

[1*]INP Clermont Auvergne, CNRS, UMR 6158 LIMOS, Univ Clermont Auvergne, Mines Saint-Etienne, 1 Rue de la Chebarde, Aubiere Cedex, 63170, Puy-de-Dome, France.

[2]LCOMS, Université de Lorraine, Metz, F-57000, France.

*Corresponding author(s). E-mail(s): viet_hung.nguyen@uca.fr; Contributing authors: thanh_loan.nguyen@doctorant.uca.fr; minh-hieu.nguyen@univ-lorraine.fr;

Abstract

In this paper, we investigate bi-objective combinatorial optimization (BOCO) problems, where two conflicting objectives must be balanced over discrete structures. While exact methods can in principle enumerate the Pareto-optimal solution set, its cardinality typically grows exponentially with the instance size, making it difficult to derive a fair and practically usable recommendation to the decision-maker. To address this challenge, researchers often seek a mathematically well-defined criterion that selects a fairness-based preferred solution. One such criterion, recently proposed in the literature, is the Nash Fairness (NF) solution based on proportional fairness [31]. However, the NF solution may fail to exist, or it can lead to multiple solutions, which limits its applicability in practice. Motivated by these limitations, we explore the Kalai-Smorodinsky (KS) solution, a classical concept from cooperative game theory, and adapt it to the discrete setting of BOCO. We introduce a novel definition of the KS solution for BOCO and prove two structural properties: a KS solution always exists, and there are at most two KS solutions. Building on these properties, we design a generic two-phase algorithm in which each phase performs a binary-search procedure over a sequence of BOCO subproblems. When these subproblems admit polynomial-time algorithms, the overall approach computes a KS solution in weakly polynomial time.

Our results provide (i) the first existence result together with tight structural bounds (at most two solutions) for the KS solution in BOCO, offering substantially greater stability than NF criterion, (ii) the first computational framework for determining KS solutions on discrete Pareto front, and (iii) an extensive experimental assessment on the Bi-objective Knapsack and Bi-objective Spanning Tree problems, demonstrating the robustness and competitiveness of the KS approach.

Keywords: Bi-Objective Combinatorial Optimization, Kalai-Smorodinsky solution, Two-Phase Method

1 Introduction

Bi-Objective Combinatorial Optimization (BOCO) concerns the simultaneous optimization of two conflicting objectives over discrete structures such as graphs, set systems, or assignment configurations. Classic examples include the Bi-Objective Spanning Tree Problem, which typically involves simultaneous minimization of two total weight-based objectives [45, 43], the Bi-Objective Assignment Problem [37], and the Bi-Objective Knapsack Problem, where two profit measures are maximized under a capacity constraint [26, 50]. Motivated by these problem types, this paper considers a general BOCO model in which each objective may be either minimized or maximized, depending on the problem context.

In contrast to single-objective optimization, a central challenge in BOCO stems from the structure of the *Pareto-optimal solutions*. A feasible solution is Pareto optimal if no other solution improves one objective without worsening the other, and the set of all such solutions, called the *Pareto front*, can be exponentially large in the size of the instance. Moreover, the Pareto optimal set typically includes both *supported* solutions, lying on the convex hull of the objective image, and *non-supported* solutions located in its interior [27]. Several exact methods have been developed to enumerate the Pareto-optimal set. The Weighted Sum Method [27] detects only supported solutions by minimizing convex combinations of the objectives. The two-phase method [49, 37] overcomes this limitation by first computing all supported solutions via scalarization in Phase I, then solving additional constrained problems to discover the non-supported solutions in Phase II. The ϵ-constraint method [11, 25] proceeds by optimizing one objective while turning the other into a constraint, thereby identifying supported and non-supported solutions. Another exact approach is the Bi-objective Branch-and-bound algorithm [36], which explores the feasible space using multi-objective bounds and systematically prunes dominated regions. While these methods provide a full representation of the Pareto front, the cardinality of this set may grow exponentially with the instance size, limiting the practicality of full enumeration and complicating subsequent decision-making.

To overcome these limitations, it is crucial to identify a mathematically well-defined criterion for selecting preferred trade-off solutions. Parametric techniques such as the budgeted constraint method [25], Ordered Weighted Averaging (OWA) operators [52], and lexicographic optimization [8] allow the decision-maker to express preferences. But the primary drawback of these methods is often highly sensitive to parameter choices that may lack a clear behavioral interpretation. Among all these techniques, fairness-based selection criteria have emerged as a principled approach to balance competing objectives. In particular, the Nash Fairness (NF) solutions for BOCO, recently studied in [31], build on the principle of proportional fairness to select solutions where the sum of proportional changes in objective values, compared to any other feasible solution, is non-negative. This framework applies when both objectives are to be minimized. A related criterion, Proportional Fairness (PF), was introduced in [32] for cases where both objectives are to be maximized. It selects solutions that proportionally balance gains and can be found using scalarization techniques. In both cases, however, the corresponding fairness-based solution set may be empty or may contain multiple solutions. This lack of existence and boundedness properties limits the robustness and interpretability of these criteria in BOCO.

A natural candidate for a more robust fairness-based selection criterion is the Kalai-Smorodinsky (KS) solution [15], originally introduced in cooperative bargaining theory. In a bargaining game, the KS solution identifies a fair compromise between two rational players by preserving proportional gains from a disagreement (reference) point to their ideal outcomes. The effectiveness of this solution concept lies in its strong theoretical foundations, such as Pareto optimality, individual rationality, and monotonicity [15, 16]. It has been employed in several continuous multi-objective optimization problems, including engineering design and process optimization [13, 28]. Let the Utopia and Nadir points represent the ideal (best) and worst feasible objective values, respectively (resp.) [5]. Then, in continuous BO, the KS solution is geometrically defined as the unique intersection point between the Pareto front and the line segment connecting the Utopia and Nadir points (Utopia-Nadir line).

However, extending this KS solution approach from continuous BO to BOCO, where the Pareto front is discrete, faces critical challenges. Specifically, in discrete settings, the Utopia-Nadir line generally does not intersect the Pareto-optimal solution set, making the direct application of the traditional KS solution infeasible. To overcome this difficulty, we propose a novel definition of the KS solution specifically for BOCO. Our method is inspired by ideas from bargaining problems with a finite set of feasible outcomes, as studied by Xu [51] and Nagahisa [29]. A preliminary attempt to introduce KS solutions in BOCO was made in [33] for the Bi-objective Spanning Tree Problem, where only supported Pareto-optimal solutions were considered. To the best of our knowledge, there is currently no KS formulation that is defined on the entire Pareto-optimal solution set of a general BOCO problem and no structural analysis of the corresponding solution set.

To bridge this gap, we introduce the concept of *KS solution* in the context of BOCO. Our definition is inspired by bargaining models with a finite set of feasible outcomes [51, 29] and is explicitly defined on the entire Pareto-optimal solution set, including both supported and non-supported Pareto-optimal solutions. We prove two structural results. First, a KS solution always exists for any BOCO instance. Second, the KS solution set contains at most two Pareto-optimal solutions. These properties provide, for the first time in the BOCO literature, existence and boundedness guarantees for a fairness-based selection criterion and sharply contrast with the behavior of the NF solutions and PF solutions. We also propose a computational framework for determining KS solutions in BOCO. Our approach adapts the classical two-phase method to the computation of KS solutions. Each phase relies on a binary search procedure that generates a sequence of BOCO subproblems. When these subproblems can be solved in polynomial time, the overall procedure computes a KS solution in weakly polynomial time. This yields the first provably efficient algorithm for computing KS solutions in BOCO, under standard assumptions on the underlying subproblems.

The contributions of this paper are threefold. (i) We propose a *general KS formulation* for BOCO that is defined on the full Pareto-optimal solution set and addresses an important gap in fairness-based multiobjective decision-making. (ii) We establish *structural guarantees* for the resulting KS solution set: for any BOCO instance, a KS solution exists and the KS solution set contains at most two Pareto-optimal solutions. (iii) We design the first *computational framework* for KS in BOCO, based on a two-phase binary-search scheme, and we show through computational experiments on the Bi-objective Knapsack Problem and the Bi-objective Spanning Tree Problem that the KS approach is robust, computationally efficient, and more stable than NF solutions and PF solutions.

The remainder of the paper is organized as follows. Section 2 formalizes the KS solution in the BOCO setting. Section 3 presents the two-phase algorithm. Section 4 reports computational experiments and comparative analyses. Section 5 concludes the paper and outlines possible directions for future research.

2 Kalai-Smorodinsky solution(s) for BOCO

2.1 Preliminaries

In this paper, we introduce the concept of Kalai-Smorodinsky (KS) solutions for BOCO problems. These solutions are regarded as preferred solutions in the Pareto set, as they provide a balanced trade-off between conflicting objectives according to a cooperative bargaining perspective. We consider a general BOCO problem in which the two objective functions may take positive or negative values. That is, each objective can be either maximized or minimized depending on the modeling context. However, without loss of generality, we reformulate the problem in a minimization setting, as follows

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

where $\mathcal{X}$ denotes the finite set of all feasible decision vectors x. For this problem, we suppose that two objectives $f_1(x), f_2(x)$ take integer values for all $x \in \mathcal{X}$. This work focuses on the objective values $(f_1(x), f_2(x))$ instead of the decision vectors x. Under this characterization, two feasible decision vectors x and x' such that $f_1(x) = f_1(x')$ and $f_2(x) = f_2(x')$ will be considered equivalent. Hence, the concept of a solution is defined in terms of the pair of objective values $(f_1(x), f_2(x))$.

A feasible solution $(f_1(x), f_2(x))$ is an *efficient solution* (i.e., *Pareto-optimal solution*) if it is non-dominated by any other solution. In other words, there exists no feasible vector $x^* \in \mathcal{X} \setminus \{x\}$ such that $f_1(x^*) \leq f_1(x)$ and $f_2(x^*) \leq f_2(x)$ with at least one strict inequality. The set of all efficient solutions forms the *Pareto front*. We recall an important concept in BOCO: the distinction between supported and non-supported efficient solutions. We formalize this notion below.

Definition 1 (Supported efficient solution) An efficient solution $(f_1(x), f_2(x))$ is called *supported* if there exists a scalar $\lambda \in [0,1]$ such that x is an optimal solution of the weightedsum problem [35]

$$\min_{x \in \mathcal{X}} (1-\lambda) f_1(x) + \lambda f_2(x).$$

Equivalently, supported efficient solutions correspond exactly to the efficient solutions located on the boundary of the convex hull of the Pareto-optimal solution set. All efficient solutions that are not supported are termed *non-supported* efficient solutions; these lie in the interior of the convex hull of the Pareto-optimal solution set.

It is worth noting that, in multi-objective linear programming, all efficient solutions are supported [47]. In contrast, BOCO problems have a discrete feasible set, so non-supported efficient solutions may appear. Their existence has been observed in several combinatorial problems. For example, in the bi-objective assignment problem, even though the constraint matrix is totally unimodular, non-supported efficient solutions still exist [49].

2.2 Definition of KS solution(s) for BOCO

Originally developed in cooperative game theory, the KS solution provides a fair compromise in bargaining problems by ensuring proportional gains for all parties from a common disagreement point. We begin by recalling the main elements of the classical bargaining framework. A bargaining problem is modeled as a pair $(\mathcal{X}, d)$, where $\mathcal{X} \subset \mathbb{R}^n$ denotes the set of feasible utility allocations that the players may jointly agree upon, and $d \in \mathbb{R}^n$ is the disagreement point that represents the outcome if no agreement is reached. The objective is to select a solution in $\mathcal{X}$ that maximizes the utilities of the players fairly and rationally. Throughout this work, we concentrate on the case $n = 2$, where a feasible allocation is denoted by $x = (x_1, x_2) \in \mathcal{X}$. By translation invariance, we may assume without loss of generality that $d = (0, 0)$. The *ideal point* $p(\mathcal{X}) = (p_1(\mathcal{X}), p_2(\mathcal{X}))$ is defined as the component-wise maximum achievable utilities for each player in the feasible set $\mathcal{X}$ namely

$$p_1(\mathcal{X}) = \max\{x_1 : (x_1, x_2) \in \mathcal{X}\}, \quad p_2(\mathcal{X}) = \max\{x_2 : (x_1, x_2) \in \mathcal{X}\}.$$

The KS solution was introduced by Kalai and Smorodinsky [15] as an alternative to the classical Nash solution [30], offering another notion of fairness by replacing the axiom of *independence of irrelevant alternatives (IIA)* with the axiom of *monotonicity*. More precisely, the KS solution satisfies the following axioms: (i) Pareto Optimality, (ii) Symmetry (or egalitarian), (iii) Invariance to affine transformations, and (iv) Monotonicity. Kalai and Smorodinsky demonstrated that a unique solution exists that satisfies all these axioms. The symmetry axiom states that if the bargaining problem treats both players identically, that is, the feasible set and the disagreement point are unchanged under a permutation of players, then the solution must assign equal utilities to both. The invariance to affine transformations requires the solution to remain unchanged under any positive affine transformation of the utility functions. The monotonicity axiom asserts that if one player's maximal attainable utility increases while the other's does not worsen, then the solution must not assign a lower utility to the improving player. Unlike the Nash solution, which relies on the IIA to ensure that the removal of unchosen feasible points does not affect the outcome, the KS solution replaces this axiom with monotonicity to promote responsiveness and fairness. In contrast, monotonicity ensures that improvements in a player's feasible outcomes are reflected in their assigned utility, without disadvantaging them when the other player's prospects remain unchanged. Under these four axioms, Kalai and Smorodinsky proved that there exists a unique solution satisfying all of them. This solution can be characterized as the unique efficient solution lying on the line segment between the disagreement point and the ideal point, where both players receive equal relative ratio of their respective maximal utilities. Formally, the KS solution is defined as follows.

Definition 2 (Kalai-Smorodinsky solution for convex game [15]) Let $\mathcal{X} \subset \mathbb{R}^2$ be a closed, convex, and comprehensive set. Let the disagreement point be $d = (0, 0)$ and the ideal point be $p(\mathcal{X}) = (p_1(\mathcal{X}), p_2(\mathcal{X}))$. The Kalai-Smorodinsky solution $x^{KS} \in \mathcal{X}$ is the unique efficient solution satisfying

$$\frac{x_1^{KS}}{p_1(\mathcal{X})} = \frac{x_2^{KS}}{p_2(\mathcal{X})}.$$

This original definition relies on the continuity of $\mathcal{X}$ to guarantee both existence and uniqueness of the solution. Geometrically, it corresponds to the point on the Pareto frontier that lies on the straight line joining the disagreement point to the ideal point. In bi-objective optimization, these two reference outcomes are often interpreted through the so-called Utopia and Nadir points, which delimit the best and worst attainable values along the efficient frontier. Although the disagreement point in the bargaining model is not identical to the Nadir point, it plays an analogous role in defining the boundary of relevant outcomes. Under this geometric interpretation, the KS solution can be viewed as the solution lying on the Utopia-Nadir line and closest to the Utopia point [27, 8].

In the multi-criteria decision-making literature, a conceptually related construction is TOPSIS (Technique for Order Preference by Similarity to an Ideal Solution), introduced in [12] and further developed in [53, 34]. TOPSIS evaluates each alternative through its Euclidean distance to an ideal and a negative-ideal point, after a normalization step applied to all criteria. Because Euclidean distance is not invariant under affine transformations of the objectives, the resulting ranking is highly sensitive to the chosen normalization scheme: simple rescaling or shifting of a single criterion may alter the closeness coefficients and hence the final ordering of solutions. This stands in clear contrast with the Kalai-Smorodinsky solution, whose definition relies on proportional gains relative to the Utopia and Nadir outcomes and is invariant under all positive affine transformations. The affine-invariance property ensures that the KS solution reacts only to genuine modifications of the feasible outcomes, rather than to artefacts introduced by scaling or normalization procedures.

Classical formulations assume that the feasible set $\mathcal{X}$ is convex. However, in a *finite game*, i.e., when the feasible set $\mathcal{X}$ is finite, the Pareto front may not intersect the line segment connecting the disagreement point and the ideal point. Thus, the standard KS concept may not yield a feasible solution [23]. To address this, several generalized definitions of the KS solution have been proposed. In particular, [29] introduced a discrete extension based on a Leontief utility function [24], which evaluates a solution by the smallest proportion of each objective achieved relative to its ideal value. Independently, [51] formulated a nonconvex extension of the KS solution that selects feasible solutions maximizing the minimum ratio of attained to ideal utility. Both approaches lead to the same reformulation in the finite case: instead of requiring the solution to lie on the line connecting the disagreement and ideal points, the KS solution is redefined as the set of feasible solutions that maximize the minimum ratio between the achieved and ideal values across all objectives. The following formulation follows the principles of both approaches.

Definition 3 (Kalai-Smorodinsky solution for finite game [29, 51]) Let $\mathcal{X} \subset \mathbb{R}^2$ be a finite feasible set, the disagreement point be $d = (0,0)$, and the ideal point be $p(\mathcal{X}) = (p_1(\mathcal{X}), p_2(\mathcal{X}))$. The Kalai-Smorodinsky solution set is given by

$$S_{KS}(\mathcal{X}) = \left\{ x^* \in \mathcal{X} \;\middle|\; \min\left(\frac{x_1^*}{p_1(\mathcal{X})}, \frac{x_2^*}{p_2(\mathcal{X})}\right) \geq \min\left(\frac{x_1}{p_1(\mathcal{X})}, \frac{x_2}{p_2(\mathcal{X})}\right),\ \forall x \in \mathcal{X} \right\}.$$

This formulation selects the solution(s) in $\mathcal{X}$ that maximize the minimum shifted utility across players, thereby preserving the relative fairness of the original KS rule while ensuring applicability in discrete structures.

This notion of KS fairness extends naturally to BOCO, where the two objectives can be interpreted as two agents negotiating over a shared feasible solution set. While the classical bargaining model is expressed in terms of utility maximization, BOCO problems are formulated in the minimization form. To adapt the KS framework to this setting, it is necessary to introduce reference points that play roles conceptually analogous to the ideal and disagreement outcomes in bargaining. In bi-objective minimization, the roles of the ideal and disagreement points are naturally played by the Utopia and Nadir points [7], resp. Their formal definitions are given below.

Definition 4 (Utopia and Nadir points [7]) Given two objective functions $f_1(x)$ and $f_2(x)$ defined over a feasible set $\mathcal{X} \subseteq \{0,1\}^n$, the *Utopia point* (f_1^U, f_2^U) is the point where both objectives resp take best attainable values

$$f_1^U = \min_{x \in \mathcal{X}} f_1(x), \quad f_2^U = \min_{x \in \mathcal{X}} f_2(x).$$

On the other hand, the *Nadir point* provides the worst objective values among the set of efficient solutions. Specifically, the Nadir point (f_1^N, f_2^N) is given by

$$f_1^N = \max_{x \in \mathcal{X}^*} f_1(x), \quad f_2^N = \max_{x \in \mathcal{X}^*} f_2(x),$$

where $\mathcal{X}^*$ denotes the set of efficient solutions.

Computing the Utopia point is straightforward, as it only requires solving two independent single-objective optimization problems. In contrast, determining the Nadir point is considerably more involved. Its computation requires complete knowledge of the efficient set $\mathcal{X}^*$, whose size may be exponential in n. Moreover, characterizing the extreme values of each objective over $\mathcal{X}^*$ is itself a nontrivial multi-objective task. As emphasized in [27, 8], no efficient general-purpose method exists for computing the exact Nadir point in multi-objective optimization, and exact computation is typically feasible only for small or highly structured instances. For this reason, a basic and widely used approximation is adopted in the BOCO literature. Let $x_1^*, x_2^* \in \mathcal{X}$ be optimal solutions of the two problems

$$x_1^* \in \operatorname*{argmin}_{x \in \mathcal{X}} f_1(x), \qquad x_2^* \in \operatorname*{argmin}_{x \in \mathcal{X}} f_2(x).$$

The Utopia point is then given by $(f_1^U, f_2^U) = \big(f_1(x_1^*),\ f_2(x_2^*)\big)$, and the Nadir point is approximated by $(f_1(x_2^*), f_2(x_1^*))$. This approximation follows from a simple and rigorous observation. Since x_2^* minimizes f_2 over $\mathcal{X}$, we have $f_2(x_2^*) \leq f_2(x)$ for all $x \in \mathcal{X}^*$. Take any efficient solution $x \in \mathcal{X}^*$. If $f_1(x) > f_1(x_2^*)$ held, then $f_1(x_2^*) < f_1(x)$ and $f_2(x_2^*) \leq f_2(x)$, so x_2^* would dominate x, contradicting the efficiency of x. Thus every efficient solution satisfies $f_1(x) \leq f_1(x_2^*)$, and hence

$$f_1^N = \max_{x \in \mathcal{X}^*} f_1(x) \leq f_1(x_2^*).$$

A symmetric argument applied to x_1^* shows that $f_2^N \leq f_2(x_1^*)$ also holds. Therefore, $(f_1(x_2^*), f_2(x_1^*))$ provides a componentwise upper bound on the exact Nadir point. In many bi-objective problems, this bound is empirically close to the actual extreme values attained on the Pareto front, which justifies its use in practice as a surrogate Nadir point [8]. Although these points play a key role in defining the solution space, the Utopia point is not a feasible solution, and the Nadir point is not an efficient one. Rather, they define the solution space for the Pareto front. These points serve as important references for normalizing objective values measured on different scales and are commonly used to guide the search for efficient solutions and to evaluate performance in BOCO problems.

Since the classical KS solution assumes that the disagreement point is at the origin, we apply a coordinate shift to the feasible set with respect to the coordinates of the Nadir and Utopia points. Specifically, for the minimization version of BOCO, the feasible set is translated such that the Utopia point is mapped to the origin and the Nadir point becomes the reference for scalarization. Let

$$\begin{aligned} P(x) &= f_1(x) - f_1(x_1^*), \\ Q(x) &= f_2(x) - f_2(x_2^*), \end{aligned}$$

for all $x \in \mathcal{X}$. Since $f_1(x)$ and $f_2(x)$ take integer values for all $x \in \mathcal{X}$, and x_1^*, x_2^* are the minimizers of f_1 and f_2, resp, it follows that $f_1(x) \geq f_1(x_1^*) \quad \text{and} \quad f_2(x) \geq f_2(x_2^*)$ for every feasible solution $x \in \mathcal{X}$. Therefore, the shifted objective values $P(x) \geq 0$ and $Q(x) \geq 0$ for all $x \in \mathcal{X}$. Moreover, since f_1 and f_2 are integer-valued, $P(x)$ and $Q(x)$ are also integers. This ensures that the shifted objectives $P(x)$ and $Q(x)$ are non-negative integers for all feasible solutions x, which allows us to consider all objective values as positive integers from this point onward. Observe that the Utopia point is now transformed to the origin of the coordinate $(0, 0)$. Furthermore, let

$$P_{\max} = f_1(x_2^*) - f_1(x_1^*) \text{ and } Q_{\max} = f_2(x_1^*) - f_2(x_2^*).$$

The Nadir point is now $(P_{\max}, Q_{\max})$ as shown in the following claim.

Claim 1. *$P_{\max}$ (resp, $Q_{\max}$) represents the maximum achievable value of objective P (resp, Q) over the set of all efficient solutions (P, Q).*

Proof Let $(f_1(x), f_2(x))$ be an arbitrary efficient solution of a BOCO problem with $x \in \mathcal{X}$. By the optimality of x_1^* and x_2^*, we obtain $f_1(x) \geq f_1(x_1^*), f_2(x) \geq f_2(x_2^*)$. Moreover, the efficiency of $(f_1(x), f_2(x))$ implies $f_2(x) \leq f_2(x_1^*)$ and $f_1(x) \leq f_1(x_2^*)$. We then deduce

$$P(x) = f_1(x) - f_1(x_1^*) \leq f_1(x_2^*) - f_1(x_1^*) = P_{\max}, \quad \text{and}$$
$$Q(x) = f_2(x) - f_2(x_2^*) \leq f_2(x_1^*) - f_2(x_2^*) = Q_{\max},$$

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 $\mathcal{D}$ represent the set of all feasible solutions (P, Q) of a BOCO problem and let $\mathcal{S} \subset \mathcal{D}$ represent the set of pairs (P, Q) corresponding to all efficient solutions. This paper characterizes efficient solutions for BOCO using pairs (P, Q) instead of explicitly listing the decision vector solutions. Thus, two efficient solutions with the same values (P, Q) will be considered equivalent. Throughout this paper, we use the notation "$\equiv$" to denote equivalent solutions.

In the context of continuous bi-objective optimization, the axiom of symmetry implies that both objectives are treated equally, and invariance to affine transformations ensures that the solution remains unchanged under scaling. These properties imply that the KS solution is the unique efficient point located at the intersection between the Pareto front and the Utopia-Nadir line. However, in the discrete setting of BOCO, such an intersection may not exist or may not be unique, and the axioms do not directly imply a geometric characterization. In this paper, we focus on BOCO problems with two objectives to be minimized, and accordingly adapt the KS definition to this setting. We now formally define the KS solution for BOCO, inspired by Definition 3.

Definition 5 (KS solution for BOCO) An efficient solution $(P^*, Q^*) \in \mathcal{S}$ is defined as a KS solution for the BOCO problem if it satisfies the following condition

$$\max\left\{\rho_{P^*}, \rho_{Q^*}\right\} \leq \max\left\{\rho_P, \rho_Q\right\}, \quad \forall (P, Q) \in \mathcal{S}, \tag{1}$$

where $\rho_P = \frac{P}{P_{\max}}$, $\rho_Q = \frac{Q}{Q_{\max}}$.

The KS solution can be seen as the best solution based on the ratio relative to the Nadir point for each objective. Geometrically, it corresponds to the efficient solution that lies closest to the Utopia-Nadir line in the shifted objective space (see Figure 1).

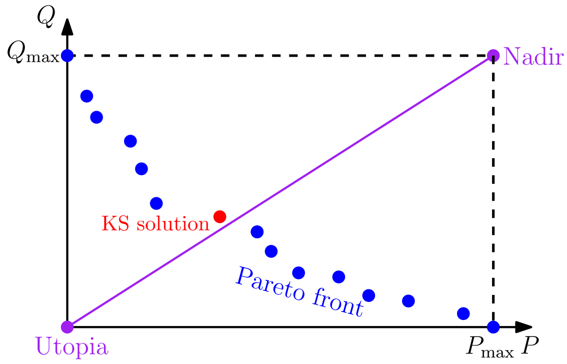


Fig. 1: Example of Utopia, Nadir points, and KS solution on the Pareto front for a BOCO problem after shifting two objectives.

When the Utopia-Nadir line intersects the Pareto front, the KS solution is unique and coincides with the intersection point. Otherwise, there may exist two KS solutions, depending on the structure of the Pareto front. We will formalize these observations and characterize the number and structure of KS solutions for BOCO in the following sections. In practice, however, BOCO problems typically yield a unique KS solution, as confirmed by our computational experiments.

2.3 Characterization of KS solution(s) for BOCO problem

The KS solution in BOCO depends fundamentally on the relative positions of efficient points with respect to the Utopia-Nadir line. To analyze this structure, we first partition the efficient solution set $\mathcal{S}$ into two subsets according to the sign of the difference between the shifted objective values ρ_P and ρ_Q introduced in Definition 5. This separation identifies efficient solutions lying on either side of the Utopia-Nadir line.

$$\mathcal{S}^+ := \{(P,Q) \in \mathcal{S} \mid \rho_P - \rho_Q \geq 0\}, \tag{2a}$$
$$\mathcal{S}^- := \{(P,Q) \in \mathcal{S} \mid \rho_P - \rho_Q \leq 0\}. \tag{2b}$$

Definition 6 Given an efficient solution (P, Q), the KS measure of (P, Q) is defined as

$$|\rho_P - \rho_Q|,$$

which quantifies the deviation of (P, Q) from the Utopia-Nadir line in the shifted objective space.

A smaller KS measure indicates a more balanced trade-off between the two objectives. If the KS measure equals zero, the corresponding point lies exactly on the Utopia-Nadir line and is the unique KS solution. In continuous bi-objective optimization, this situation always occurs, as the Pareto front necessarily intersects the Utopia-Nadir line, implying that the KS solution has KS measure equal to zero. In contrast, for BOCO problems, the efficient solution set is discrete, and this intersection often fails to exist; consequently, the KS measure of the KS solution typically remains strictly positive. To identify the most balanced trade-off efficient solutions on each side of the line, we select the minimizers of the KS measure within $\mathcal{S}^+$ and $\mathcal{S}^-$.

Definition 7 The efficient solutions that minimize the KS measure in the sets $\mathcal{S}^+$ and $\mathcal{S}^-$ are denoted as (P^+, Q^+) and (P^-, Q^-), resp. Formally, they are defined as follows

$$(P^+, Q^+) = \underset{(P,Q)\in\mathcal{S}^+}{\operatorname{argmin}} \, (\rho_P - \rho_Q), \tag{3a}$$

$$(P^-, Q^-) = \underset{(P,Q)\in\mathcal{S}^-}{\operatorname{argmax}} \, (\rho_P - \rho_Q). \tag{3b}$$

These two points represent the efficient solutions lying closest to the Utopia-Nadir line when distances are evaluated in the shifted and range-scaled objective values (ρ_P, ρ_Q). Figure 2 illustrates this geometric separation.

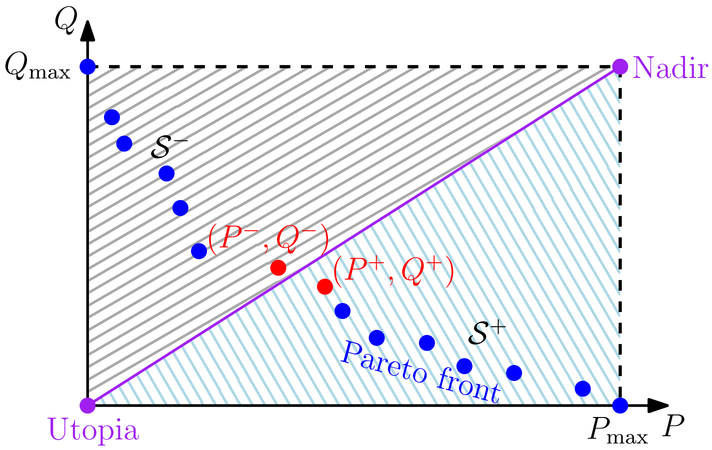


Fig. 2: Illustration of (P^+, Q^+) and (P^-, Q^-) as the closest efficient solutions to the Utopia-Nadir line in $\mathcal{S}^+$ and $\mathcal{S}^-$, resp.

The next lemma establishes a monotonicity property that characterizes these two solutions.

Lemma 1. *Let (P^+, Q^+) and (P^-, Q^-) be defined as in (3a) and (3b). Then*

(i) an efficient solution $(P^*, Q^*) \in \mathcal{S}^+$ satisfies $(P^*, Q^*) \equiv (P^+, Q^+)$ if and only if $\forall\,(P,Q) \in \mathcal{S}^+$, $P^* \leq P$.

(ii) an efficient solution $(P^*, Q^*) \in \mathcal{S}^-$ satisfies $(P^*, Q^*) \equiv (P^-, Q^-)$ if and only if $\forall\,(P,Q) \in \mathcal{S}^-$, $Q^* \leq Q$.

Proof By symmetry, it suffices to prove (i).

($\Rightarrow$) Let (P, Q) be an efficient solution that belongs to $\mathcal{S}^+$. From (3a), we have

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

Since both (P^+, Q^+) and (P, Q) are efficient solutions, it follows that $P^+ \leq P$; otherwise, if $P^+ > P$ and $Q^+ < Q$, then $\rho_{P^+} - \rho_{Q^+} = P^+/P_{\max} - Q^+/Q_{\max} > P/P_{\max} - Q/Q_{\max} = \rho_P - \rho_Q$, leading to a contradiction.

($\Leftarrow$) Conversely, assume that $(P^*, Q^*) \in \mathcal{S}^+$ is an efficient solution such that $P^* \leq P$ for all $(P, Q) \in \mathcal{S}^+$. Since all solutions in $\mathcal{S}^+$ are efficient, we have $Q^* \geq Q$. It follows that

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

Therefore, (P^*, Q^*) attains the minimum in (3a), proving $(P^*, Q^*) \equiv (P^+, Q^+)$. □

The next lemma shows that (P^+, Q^+) and (P^-, Q^-) lie consecutively on the Pareto front and constitute the only efficient points that are closest to the Utopia-Nadir line on their respective sides.

Lemma 2. *Let (P^+, Q^+) and (P^-, Q^-) be defined as in (3a) and (3b). If $(P^+, Q^+) \neq (P^-, Q^-)$, then these two points are unique consecutive supported efficient solutions located on opposite sides of the Utopia-Nadir line.*

Proof Uniqueness in each subset follows directly from Lemma 1. Suppose, for contradiction, that an efficient solution (P, Q) satisfies

$$P^- < P < P^+.$$

Efficiency of (P^+, Q^+) and (P^-, Q^-) then forces

$$Q^- > Q > Q^+.$$

If $(P, Q) \in \mathcal{S}^+$, Lemma 1(i) would require $P \geq P^+$, contradicting $P < P^+$. If $(P, Q) \in \mathcal{S}^-$, Lemma 1(ii) would require $Q \geq Q^-$, contradicting $Q < Q^-$. Thus, no efficient solution lies between them, establishing consecutiveness. □

The following theorem establishes the relationship between the KS solution(s) and the efficient solutions that achieve the minimum KS measure in the sets of efficient solutions separated by the Utopia-Nadir line. In particular, it also shows that the KS solution always exists for any BOCO problem and admits at most two solutions. The theorem precisely characterizes the conditions under which the KS solution is unique or when two distinct solutions occur.

Theorem 1 *Let (P^+, Q^+) and (P^-, Q^-) be the efficient solutions defined by (3a) and (3b), resp. For a given BOCO problem, the KS solution(s) are characterized as follows*

$$\textit{KS solution(s)} = \begin{cases} (P^+, Q^+) \text{ and } (P^-, Q^-) & \text{if } \rho_{P^+} = \rho_{Q^-}, \\ (P^+, Q^+), & \text{if } \rho_{P^+} < \rho_{Q^-}, \\ (P^-, Q^-), & \text{otherwise.} \end{cases}$$

Proof Let (P, Q) be an efficient solution that belongs to $\mathcal{S}^+$. Due to lemma 1, we get $P^+ \leq P$, it follows that

$$\rho_{P^+} \leq \rho_P, \quad \forall (P, Q) \in \mathcal{S}^+.$$

Similarly, for $\mathcal{S}^-$, we deduce that $\rho_{Q^-} \leq \rho_Q$. We then obtain the following inequalities:

$$\max\{\rho_{P^+}, \rho_{Q^+}\} = \rho_{P^+} \leq \rho_P = \max\{\rho_P, \rho_Q\}, \quad \forall (P,Q) \in \mathcal{S}^+, \tag{follow 2a}$$

$$\max\{\rho_{P^-}, \rho_{Q^-}\} = \rho_{Q^-} \leq \rho_Q = \max\{\rho_P, \rho_Q\}, \quad \forall (P,Q) \in \mathcal{S}^-. \tag{follow 2b}$$

Hence, if $\rho_{P^+} < \rho_{Q^-}$ (resp. $\rho_{Q^-} < \rho_{P^+}$), then $\max\{\rho_{P^+}, \rho_{Q^+}\} \leq \max\{\rho_P, \rho_Q\}, \forall (P,Q) \in \mathcal{S}$. This means that (P^+, Q^+) (resp. (P^-, Q^-)) is the unique KS solution. Otherwise, the two values coincide, and the KS solution set includes both (P^+, Q^+) and (P^-, Q^-). □

If $(P^+, Q^+) \equiv (P^-, Q^-)$, then the KS solution (P^*, Q^*) is precisely the unique intersection of the Pareto front and the Utopia-Nadir line. This coincides with the continuous cases, in which $(P^*, Q^*) \equiv (P^+, Q^+) \equiv (P^-, Q^-)$. Otherwise, these points correspond to the two consecutive supported efficient solutions located on opposite sides of the Utopia-Nadir line. By Theorem 1, the KS solution(s) must lie among these two solutions, which implies there is always at least one KS solution and at most two distinct KS solutions. Although this theoretical upper bound allows for two KS solutions, such cases rarely occur in practice. This observation will be further supported and discussed through the numerical experiments in Section 4. The remainder of the paper focuses on the determination of (P^+, Q^+) and (P^-, Q^-), which are critical for characterizing the KS solution.

3 A two-phase method for finding KS solution(s)

In this section, we focus on computing the KS solution(s) for a BOCO problem. As established in Section 2.3, the KS solution(s) must belong to the pair of consecutive efficient solutions (P^+, Q^+) and (P^-, Q^-), which lie on opposite sides of the Utopia-Nadir line. Therefore, the main objective of this section is to determine these two solutions.

To compute efficient solutions in BOCO problems, several procedures are available in the literature. In this work, we rely on the two-phase method, a well-established and widely adopted approach originally proposed by Ulungu and Teghem [49]. This method has been successfully applied to a variety of bi-objective integer programming problems, including the assignment problem [37], network flow [20, 44], knapsack [50], spanning tree [40, 2], and the traveling salesman problem with profits [10]. It has also been extended to multi-objective integer programs with more than two objectives [38]. In the original two-phase method, Phase I identifies all supported efficient solutions, and Phase II generates non-supported efficient solutions by restricting the search to rectangular regions between consecutive supported solutions in the objective space.

In order to identify the two consecutive efficient solutions (P^+, Q^+) and (P^-, Q^-) lying on opposite sides of the Utopia-Nadir line, we adapt the two-phase method as follows:

- Phase I focuses on identifying the two consecutive supported efficient solutions located on opposite sides of the Utopia-Nadir line.

- Phase II aims at determining, if necessary, the two consecutive non-supported efficient solutions located on opposite sides of the Utopia-Nadir line. Information from the supported efficient solutions obtained in Phase I is used to reduce the search space in Phase II.

This section introduces two binary search procedures, one for each phase of the method, and analyzes the computational complexity of the resulting algorithm.

3.1 Phase I: Identifying the two consecutive supported efficient solutions located on the opposite sides of the Utopia-Nadir line

We first outline the algorithm used in Phase I to determine the two consecutive supported efficient solutions. Let $\mathcal{S}_{SE} \subseteq \mathcal{S}$ denote the set of all supported efficient solutions of a BOCO problem. Since $\mathcal{S}$ is finite and discrete, $\mathcal{S}_{SE}$ is also finite and discrete. As a subset of the Pareto front, $\mathcal{S}_{SE}$ can be partitioned into two subsets separated by the Utopia-Nadir line: $\mathcal{S}^+_{SE} \subseteq \mathcal{S}^+$ and $\mathcal{S}^-_{SE} \subseteq \mathcal{S}^-$. Let $\mathcal{C}_{SE}$ denote the set comprising the two consecutive supported efficient solutions positioned on opposite sides of the Utopia-Nadir line.

We recall that each supported efficient solution is necessarily a solution of a weighted sum single-objective optimization problem [35]

$$\min_{(P,Q)\in\mathcal{D}} \quad \mathcal{F}(\alpha) = \ (1-\alpha)P + \alpha Q$$

where $\alpha \in [0,1]$ is a given coefficient. 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 $\mathcal{F}(\alpha_i), \mathcal{F}(\alpha_j)$, resp., we present Procedure *SEARCH_Phase_One* to find the set $\mathcal{C}_{SE}$ which can be determine by solving $\mathcal{F}(\alpha)$ with some $\alpha \in [\alpha_i, \alpha_j]$. At each iteration, we compute the midpoint α_m of the current interval and solve $\mathcal{F}(\alpha_m)$ to obtain a solution (P_m, Q_m). We then determine whether (P_m, Q_m) belongs to $\mathcal{S}^+_{SE}$ or $\mathcal{S}^-_{SE}$, and retain only the half-interval in which the two endpoint solutions lie on opposite sides of the Utopia-Nadir line. This guarantees that, after each iteration, the two endpoints of the remaining interval correspond to supported efficient solutions located on opposite sides of the Utopia-Nadir line. The process is repeated until the length of the interval becomes smaller than a positive parameter γ. The choice of γ ensures that the two endpoint solutions are consecutive supported efficient solutions on opposite sides of the Utopia-Nadir line, and the procedure therefore converges in a logarithmic number of iterations with respect to γ.

We first state a monotonicity property describing the relationship between the parameter α and the optimal solution of $\mathcal{F}(\alpha)$.

Lemma 3. *Given $0 \le \alpha_i < \alpha_j \le 1$. 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, $P_i \le P_j$ and $Q_i \ge Q_j$.*

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

$$(1-\alpha_i)P_i + \alpha_i Q_i \le (1-\alpha_i)P_j + \alpha_i Q_j, \text{ and} \tag{5a}$$
$$(1-\alpha_j)P_j + \alpha_j Q_j \le (1-\alpha_j)P_i + \alpha_j Q_i \tag{5b}$$

Multiplying both sides of (5a) by $1-\alpha_j \ge 0$ and (5b) by $1-\alpha_i \ge 0$ then adding them, we obtain $(\alpha_i - \alpha_j)(Q_i - Q_j) \le 0$. As $\alpha_i < \alpha_j$, this implies $Q_i \ge Q_j$. Moreover, from (5a), we have $(1-\alpha_i)(P_i - P_j) \le \alpha_i(Q_j - Q_i) \le 0$. As $\alpha_i < 1$, we deduce $P_i \le P_j$. □

Due to Lemma 3, 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 then present a method for determining elements in the set $\mathcal{C}_{SE}$ that are found by solving $\mathcal{F}(\alpha_i)$ and $\mathcal{F}(\alpha_j)$. The key is to determine the length of $[\alpha_i, \alpha_j]$, which plays a role in the stopping condition for the iterative process, as formalized in the following theorem.

Theorem 2 *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 $0 < \alpha_j - \alpha_i < \gamma = 4/(P_{\max} + Q_{\max})^2$. 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 $(P_s, Q_s) \equiv (P_j, Q_j)$, $\mathcal{C}_{SE} = \{(P_i, Q_i), (P_j, Q_j)\}$;*
2. *Otherwise, $\mathcal{C}_{SE} = \{(P_i, Q_i), (P_s, Q_s)\}$ or $\mathcal{C}_{SE} = \{(P_s, Q_s), (P_j, Q_j)\}$.*

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 3. 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

$$(1-\alpha_i)P_i + \alpha_i Q_i \le (1-\alpha_i)P_j + \alpha_i Q_j, \tag{6a}$$
$$(1-\alpha_j)P_j + \alpha_j Q_j \le (1-\alpha_j)P_i + \alpha_j Q_i, \tag{6b}$$

From (6a) and (6b), we get

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

which implies $\alpha_i \le \alpha_s \le \alpha_j$. We consider following two 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 3, (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, $\mathcal{C}_{SE} = \{(P_i, Q_i), (P_j, Q_j)\}$.
2. $(P_s, Q_s) \not\equiv (P_i, Q_i)$ and $(P_s, Q_s) \not\equiv (P_j, Q_j)$
   By the same argument, we also get $P_i < P_s < P_j, Q_i > Q_s > Q_j$ and

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

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

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

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, $\mathcal{C}_{SE} = \{(P_i, Q_i), (P_s, Q_s)\}$ or $\mathcal{C}_{SE} = \{(P_s, Q_s), (P_j, Q_j)\}$.

□

Because of Lemma 3, 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 2. Therefore, as a result of Theorem 2, 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 $\mathcal{S}^+_{SE}$. The following Procedure SEARCH_Phase_One is designed to find the two consecutive supported efficient solutions belonging to $\mathcal{S}^-_{SE}$

and $\mathcal{S}_{SE}^{+}$, i.e. the set $\mathcal{C}_{SE}$, obtained by solving the weighted sum problem $\mathcal{F}(\alpha)$ with $\alpha \in [0, 1]$.

Algorithm 1 *SEARCH_Phase_One*: Determine the set $\mathcal{C}_{SE}$ for a BOCO problem

Input: A BOCO problem with the Nadir point $(P_{\max}, Q_{\max})$ and the Utopia point $(0, 0)$.
Output: The set $\mathcal{C}_{SE}$.

Solve $\mathcal{F}(0), \mathcal{F}(1)$ to obtain $(P_0, Q_0), (P_1, Q_1)$, resp.
$\gamma \leftarrow 4/(P_{\max} + Q_{\max})^2$
$0 \leftarrow \alpha_i, 1 \leftarrow \alpha_j$
$(P_i, Q_i) \leftarrow (P_0, Q_0), (P_j, Q_j) \leftarrow (P_1, Q_1)$
while $\alpha_j - \alpha_i \geq \gamma$ **do**
    $\alpha_m \leftarrow (\alpha_i + \alpha_j)/2$
    Solve $\mathcal{F}(\alpha_m)$ to obtain (P_m, Q_m)
    $\rho_m = P_m/P_{\max} - Q_m/Q_{\max}$
    if $\rho_m == 0$ **then** return (P_m, Q_m)
    else if $\rho_m > 0$ **then**
        $\alpha_j \leftarrow \alpha_m, (P_j, Q_j) \leftarrow (P_m, Q_m)$
    else
        $\alpha_i \leftarrow \alpha_m, (P_i, Q_i) \leftarrow (P_m, Q_m)$
    end if
end while
$\alpha_s \leftarrow (P_j - P_i)/(P_j - P_i + Q_i - Q_j)$
Solve $\mathcal{F}(\alpha_s)$ to obtain (P_s, Q_s)
$\rho_s = P_s/P_{\max} - Q_s/Q_{\max}$
if $\rho_s == 0$ **then**
    return (P_s, Q_s)
else if $\rho_s > 0$ **then**
    return $(P_i, Q_i), (P_s, Q_s)$
else
    return $(P_s, Q_s), (P_j, Q_j)$
end if

The correctness and complexity of Algorithm 1 will be discussed at the end of this section.

3.2 Phase II: Identifying the two non-supported consecutive efficient solutions located on opposite sides of the Utopia-Nadir line

To initiate Phase II, without loss of generality, we assume that Phase I has identified two distinct supported efficient solutions located on opposite sides of the Utopia-Nadir line. We denote these solutions by $(P_s^-, Q_s^-) \in \mathcal{S}^-$ and $(P_s^+, Q_s^+) \in \mathcal{S}^+$. We define the region $\mathcal{H}$ as the interior of the rectangular area where (P_s^-, Q_s^-) forms the upper-left corner and (P_s^+, Q_s^+) forms the lower-right corner. Following Theorem 1, the objective of Phase II is to determine the solutions (P^+, Q^+) and (P^-, Q^-) that lie in the region $\mathcal{H}$. Owing to the symmetry between (P^+, Q^+) and (P^-, Q^-) with respect to the Utopia-Nadir line, this subsection focuses only on identifying the efficient solution (P^+, Q^+).

The weighted sum method used in Phase I has a significant drawback: it cannot identify non-supported efficient solutions. As stated in Lemma 1 and further clarified in Theorem 1, the solution (P^+, Q^+) is the efficient solution that minimizes P over the region $\mathcal{S}^+$ determined by the Utopia-Nadir line, and this solution may be a non-supported part of the Pareto front. Therefore, although several methods are capable of identifying non-supported efficient solutions, the ϵ-constraint method proposed by Miettinen [27] is particularly well suited to this task. This method reformulates the problem by maintaining one objective as a scalar-valued function while transforming the remaining objectives into constraints, which are bounded above by a parameter ϵ. Specifically, we consider the following ϵ-constraint problem

$$\begin{aligned} \min_{(P,Q)\in\mathcal{D}} \quad & P \\ \text{s.t} \quad & Q \le \epsilon. \end{aligned} \tag{$\mathcal{P}'(\epsilon)$}$$

where $\epsilon \ge 0$ is a parameter limited to the feasible region of Q. While the ϵ-constraint method is effective for this purpose, it has a key limitation: it does not always guarantee that the solution is efficient. To address this issue, we consider a variant of the original ϵ-constraint problem by incorporating a weighted sum of the original objectives into its formulation, ensuring the obtained solution is efficient. The revised ϵ-constraint problem is expressed as

$$\begin{aligned} \min_{(P,Q)\in\mathcal{D}} \quad & P + \frac{Q}{Q_{\max}} \\ \text{s.t} \quad & Q \le \epsilon. \end{aligned} \tag{$\mathcal{P}(\epsilon)$}$$

where $\epsilon \in [0, Q_{\max})$ is a given parameter and $Q_{\max}$ is the maximum value of objective Q for all efficient solutions (P, Q), as defined in Section 2.2. The relationship between this revised problem and the original ϵ-constraint problem is detailed in the subsequent theorem.

Theorem 3 *Given $\epsilon \in [0, Q_{\max})$, assume that (P^*, Q^*) solves the revised ϵ-constraint problem $\mathcal{P}(\epsilon)$. Then (P^*, Q^*) is an efficient solution. Moreover, (P^*, Q^*) also solves the corresponding original ϵ-constraint problem $\mathcal{P}'(\epsilon)$.*

Proof Let (P^*, Q^*) be an optimal solution to $\mathcal{P}(\epsilon)$ with $Q_{\max} > \epsilon \ge 0$. We need to prove that (P^*, Q^*) is efficient. For the sake of contradiction, suppose that (P^*, Q^*) is not efficient. Then there exists a feasible solution (P, Q) such that $P \le P^*$ and $Q \le Q^*$, with at least one of these inequalities being strict. Since (P^*, Q^*) is feasible for $\mathcal{P}(\epsilon)$, we get $Q \le Q^* \le \epsilon$. It follows that (P, Q) is also feasible for $\mathcal{P}(\epsilon)$. From the strict inequality in either P or Q, we deduce

$$P + \frac{Q}{Q_{\max}} < P^* + \frac{Q^*}{Q_{\max}},$$

contradicting the optimality of (P^*, Q^*) for $\mathcal{P}(\epsilon)$. Thus, (P^*, Q^*) must be efficient.

We now demonstrate that (P^*, Q^*) is a solution of $\mathcal{P}'(\epsilon)$. Considering any feasible solution (P', Q') for $\mathcal{P}'(\epsilon)$ satisfies $Q' \leq \epsilon$ and thus it is also feasible for $\mathcal{P}(\epsilon)$. Therefore, by the optimality in $\mathcal{P}(\epsilon)$, we have $P^* + \frac{Q^*}{Q_{\max}} \leq P' + \frac{Q'}{Q_{\max}}$. If $P^* > P'$ then we can deduce that $Q^* \leq Q'$. Using the positivity arguments as before, we get

$$P^* - P' \geq 1 > \frac{Q'}{Q_{\max}} \geq \frac{Q' - Q^*}{Q_{\max}},$$

which leads to $P^* + \frac{Q^*}{Q_{\max}} > P' + \frac{Q'}{Q_{\max}}$, a contradiction. Hence, we obtain $P^* \leq P$. In the other hand, since $Q^* \leq \epsilon$, (P^*, Q^*) is also feasible for $\mathcal{P}'(\epsilon)$. Thus, by the optimality in $\mathcal{P}(\epsilon)$, we get $P \leq P^*$. Consequently, $P^* = P$ implies that (P^*, Q^*) is also optimal for $\mathcal{P}'(\epsilon)$. $\square$

Furthermore, it is straightforward to verify that if (P^*, Q^*) is an efficient solution of $\mathcal{P}(\epsilon)$, then it is unique. Indeed, suppose there exists a solution (P_1, Q_1) of $\mathcal{P}(\epsilon)$ with $(P_1, Q_1) \neq (P^*, Q^*)$. From Theorem 3, we get $P^* = P_1 = \min\{P \mid Q \leq \epsilon\}$. Since both (P^*, Q^*) and (P_1, Q_1) are efficient solutions, we deduce that $Q^* = Q_1$, this is a contradiction.

The lemma below shows the monotonic relationship between the value ϵ and the solution (P, Q) of $\mathcal{P}(\epsilon)$.

Lemma 4. *Let (P_i, Q_i) and (P_j, Q_j) be the solutions of the problems $\mathcal{P}(\epsilon_i)$ and $\mathcal{P}(\epsilon_j)$ resp. If $\epsilon_i < \epsilon_j$ then $P_i \geq P_j$ and $Q_i \leq Q_j$.*

Proof Since (P_i, Q_i) is a solution to $\mathcal{P}(\epsilon_i)$, it satisfies the inequality $Q_i \leq \epsilon_i < \epsilon_j$. This establishes that (P_i, Q_i) is a feasible solution for $\mathcal{P}'(\epsilon_j)$. Moreover, by Theorem 3, (P_j, Q_j) is an optimal solution to $\mathcal{P}'(\epsilon_j)$, it follows that $P_j \leq P_i$. Additionally, as stated in Theorem 3, (P_i, Q_i) and (P_j, Q_j) are both efficient solutions. Therefore, we get $Q_i \leq Q_j$. $\square$

Theorem 4 *Let (P^*, Q^*) be an efficient solution to a BOCO problem. Then there exists a value $\epsilon \in [0, Q_{\max}]$ for which (P^*, Q^*) is a solution to $\mathcal{P}(\epsilon)$.*

Proof Suppose (P^*, Q^*) is an efficient solution. We then show that (P^*, Q^*) is a solution to a problem $\mathcal{P}(Q^*)$. To this end, we consider any feasible solution $(P, Q) \not\equiv (P^*, Q^*)$ of $\mathcal{P}(Q^*)$. This means $Q \leq Q^*$. Since (P^*, Q^*) is efficient, it follows that $P > P^*$. Because P, Q, P^*, Q^* are positive integers, we get

$$P - P^* \geq 1 \geq \frac{Q^*}{Q_{\max}} \geq \frac{Q^* - Q}{Q_{\max}}.$$

Rearranging, we obtain $P^* + \frac{Q^*}{Q_{\max}} \leq P + \frac{Q}{Q_{\max}}$. This shows that (P^*, Q^*) minimizes the objective $P + \frac{Q}{Q_{\max}}$ in $\mathcal{P}(Q^*)$. Hence, for $\epsilon = Q^*$, (P^*, Q^*) is optimal in $\mathcal{P}(\epsilon)$.

Lastly, since $\epsilon = Q^*$ and $Q^* \in [0, Q_{\max}]$, it follows that $\epsilon \in [0, Q_{\max}]$. $\square$

Algorithm 2 *SEARCH_Phase_Two_*$\mathcal{S}^+$: Determine the solution (P^+, Q^+)

Input: A BOCO problem.
Parameter: $\mathcal{C}_{SE} = \{(P_s^-, Q_s^-), (P_s^+, Q_s^+)\}$, the Nadir point $(P_{\max}, Q_{\max})$ and the Utopia point $(0, 0)$.
Output: The efficient solution (P^+, Q^+).

$(P^+, Q^+) \leftarrow (P_s^+, Q_s^+)$
$\epsilon_i \leftarrow Q_s^+$
$\epsilon_j \leftarrow Q_s^-$
while $\epsilon_j - \epsilon_i \geq 1$ **do**
  $\epsilon_m \leftarrow \lfloor(\epsilon_i + \epsilon_j)/2\rfloor$
  Solve $\mathcal{P}(\epsilon_m)$ to obtain a solution (P_m, Q_m)
  $\rho_m = P_m/P_{\max} - Q_m/Q_{\max}$
  if $\rho_m > 0$ **then**
    $(P^+, Q^+) \leftarrow (P_m, Q_m)$
    $\epsilon_i \leftarrow \epsilon_m$
  else if $\rho_m < 0$ **then**
    $\epsilon_j \leftarrow Q_m$
  else
    return (P_m, Q_m)
  end if
end while
return (P^+, Q^+)

Theorems 3 and 4, together with the uniqueness of the optimal solution of $\mathcal{P}(\epsilon)$, imply that for any efficient solution $(P^+, Q^+) \in \mathcal{S}^+$ minimizing P over $\mathcal{S}^+$, there exists a unique parameter $\epsilon^+ \in [Q_s^+, Q_s^-]$ such that (P^+, Q^+) is the optimal solution of $\mathcal{P}(\epsilon^+)$. This observation motivates a binary search over the integer interval $[Q_s^+, Q_s^-]$ to locate ϵ^+.

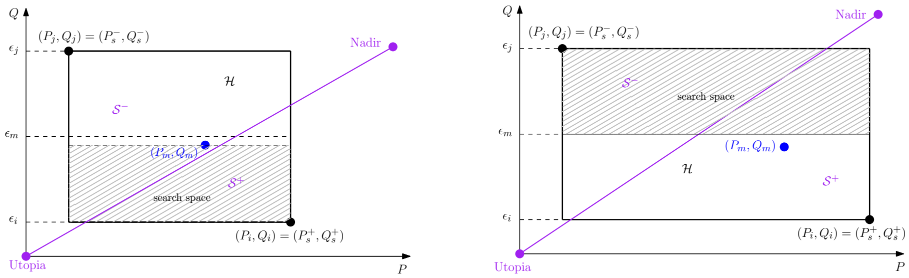
We now introduce the algorithm *SEARCH_Phase_Two_*$\mathcal{S}^+$ to find the solution (P^+, Q^+) in the set $\mathcal{S}^+$ that minimizes P among all efficient solutions $(P, Q) \in \mathcal{S}^+$. The algorithm starts by setting $\epsilon_i = Q_s^+$ and $\epsilon_j = Q_s^-$. The points (P_s^+, Q_s^+) and (P_s^-, Q_s^-) are, resp, obtained as optimal solutions of $\mathcal{P}(\epsilon_i)$ and $\mathcal{P}(\epsilon_j)$. Hence, the initial candidate solution (P^+, Q^+) is set to (P_s^+, Q_s^+). At each iteration, the midpoint of the current interval is computed and rounded down to the nearest integer

$$\epsilon_m = \left\lfloor \frac{\epsilon_i + \epsilon_j}{2} \right\rfloor,$$

where $\lfloor\cdot\rfloor$ denotes the floor function. We then solve $\mathcal{P}(\epsilon_m)$ to obtain a solution (P_m, Q_m) and determine whether (P_m, Q_m) belongs to $\mathcal{S}^+$ or $\mathcal{S}^-$, based on the sign of $\rho_m = P_m/P_{\max} - Q_m/Q_{\max}$. The search proceeds as follows.

- If $(P_m, Q_m) \in \mathcal{S}^+$ (equivalently, $\rho_m > 0$), then $\epsilon_i < \epsilon_m$ and Lemma 4 implies $P_m \leq P^+$. We update the candidate solution (P^+, Q^+) to (P_m, Q_m) (see Figure 3b), and retain the sub-interval $[\epsilon_m, \epsilon_j]$ as the new search interval.

- If $(P_m, Q_m) \in \mathcal{S}^-$ (equivalently, $\rho_m < 0$), then the sub-interval $[\epsilon_m, \epsilon_j]$ can be discarded and the search continues in $[\epsilon_i, \epsilon_m]$. Since $Q_m \leq \epsilon_m$, the interval $[Q_m, \epsilon_j]$ covers at least half of $[\epsilon_i, \epsilon_j]$. Thus, it is sufficient to retain the reduced interval $[\epsilon_i, Q_m]$ as the search interval, while keeping the current candidate solution (P^+, Q^+) unchanged (see Figure 3a).



(a) The case $(P_m, Q_m) \in \mathcal{S}^-$ (b) The case $(P_m, Q_m) \in \mathcal{S}^+$

Fig. 3: The searching solution space at the first step of SEARCH_Phase_Two_$\mathcal{S}^+$

This process ensures that each iteration reduces the search interval by half. Since P and Q are integer values, the process will stop when the length of the interval is less than 1. As a result, the algorithm converges in a logarithmic number of iterations with respect to the initial length of the interval $[Q_s^+, Q_s^-]$.

3.3 Proof of correctness and complexity

The theorem below presents the formal statement for determining the set $\mathcal{C}_{SE}$ and convergence proofs of Algorithm 1 in the first phase.

Theorem 5 *Procedure SEARCH_Phase_One determines the set $\mathcal{C}_{SE}$ in $\mathcal{O}(\log_2 (P_{\max} + Q_{\max}))$ iterations.*

Proof Note that, by construction, the shifted formulation ensures that the solution of $\mathcal{F}(0)$ is $(P_0, Q_0) = (0, Q_0) \in \mathcal{S}^-_{SE}$ and the solution of $\mathcal{F}(1)$ is $(P_1, Q_1) = (P_1, 0) \in \mathcal{S}^+_{SE}$. Thus, the initial interval $[0, 1]$ is bounded by two supported efficient solutions lying on opposite sides of the Utopia-Nadir line. Procedure SEARCH_Phase_One then applies a binary search on this interval: at each iteration, the midpoint $\alpha_m = (\alpha_i + \alpha_j)/2$ is computed, and the corresponding solution (P_m, Q_m) of $\mathcal{F}(\alpha_m)$ is obtained. Depending on whether (P_m, Q_m) belongs to $\mathcal{S}^-_{SE}$ or $\mathcal{S}^+_{SE}$, one of the subintervals $[\alpha_i, \alpha_m]$ or $[\alpha_m, \alpha_j]$ is retained, guaranteeing that the two endpoints of the new interval correspond to supported efficient solutions located on opposite sides of the Utopia-Nadir line. This invariant is preserved throughout the procedure.

The algorithm terminates when either a solution satisfying $\rho_m = 0$ is obtained (which directly yields the KS solution) or when the length of the current interval falls below the value

γ. In the latter case, by Theorem 2, the two supported efficient solutions at the endpoints (or one endpoint together with an intermediate solution) form the set $\mathcal{C}_{SE}$.

Since the interval length halves with each iteration, after k iterations, the interval length is $(1/2)^k$. We require this length to be less than or equal to γ: $\left(\frac{1}{2}\right)^k \leq \gamma \implies k \geq \log_2\left(\frac{1}{\gamma}\right)$. Hence the number of iterations is at most

$$\left\lceil \log_2\left(\frac{1}{\gamma}\right)\right\rceil = \left\lceil \log_2\left(\frac{(P_{\max}+Q_{\max})^2}{4}\right)\right\rceil = \lceil 2\log_2(P_{\max}+Q_{\max}) - 2\rceil .$$

□

Theorem 6 *Algorithm SEARCH_Phase_Two_$\mathcal{S}^+$ computes the efficient solution (P^+, Q^+) in $\mathcal{O}(\log_2 Q_{\max})$ iterations.*

Proof By construction, Phase I provides two efficient solutions $(P_s^+, Q_s^+) \in \mathcal{S}^+$ and $(P_s^-, Q_s^-) \in \mathcal{S}^-$ lying on opposite sides of the Utopia-Nadir line, with $Q_s^+ \leq Q_s^-$. Hence the initial interval $[Q_s^+, Q_s^-]$ contains a value ϵ^+ such that solving $\mathcal{P}(\epsilon^+)$ yields the solution (P^+, Q^+) that minimizes P over $\mathcal{S}^+$. Procedure SEARCH_Phase_Two_$\mathcal{S}^+$ applies a binary search on this integer interval. At each iteration, let $\epsilon_m = \lfloor(\epsilon_i + \epsilon_j)/2\rfloor$ and (P_m, Q_m) be the solution of $\mathcal{P}(\epsilon_m)$.

- If $\rho_m > 0$, then $(P_m, Q_m) \in \mathcal{S}^+$. Hence (P_m, Q_m) and (P^+, Q^+) lie on the same side of the Utopia-Nadir line. Since $\epsilon_i < \epsilon_m$, Lemma 4 yields $P_m \leq P^+$. Update $(P^+, Q^+) \leftarrow (P_m, Q_m)$ and $\epsilon_i \leftarrow \epsilon_m$, and continue on the interval $[\epsilon_i, \epsilon_j]$.
- If $\rho_m < 0$, then $(P_m, Q_m) \in \mathcal{S}^-$. Thus (P_m, Q_m) and (P^+, Q^+) lie on opposite sides of the Utopia-Nadir line, so (P^+, Q^+) is left unchanged. Moreover, since $Q_m \leq \epsilon_m$, the point (P_m, Q_m) remains feasible for every problem $\mathcal{P}(\epsilon')$ with $\epsilon' \in [Q_m, \epsilon_m]$. If there existed a solution $(\bar{P}, \bar{Q})$ with

$$\bar{P} + \frac{\bar{Q}}{Q_{\max}} < P_m + \frac{Q_m}{Q_{\max}}$$

  for some $\epsilon' \in [Q_m, \epsilon_m]$, then $(\bar{P}, \bar{Q})$ would also be feasible for $\mathcal{P}(\epsilon_m)$, contradicting the optimality of (P_m, Q_m) for $\mathcal{P}(\epsilon_m)$. Hence (P_m, Q_m) is the optimal solution of $\mathcal{P}(\epsilon')$ for all $\epsilon' \in [Q_m, \epsilon_m]$. We therefore set $\epsilon_j \leftarrow Q_m$ and continue on the reduced interval $[\epsilon_i, \epsilon_j] = [\epsilon_i, Q_m]$.
- If $\rho_m = 0$, then (P_m, Q_m) lies on the Utopia-Nadir line and the algorithm terminates with (P_m, Q_m).

This process maintains that the value of ϵ^+ always belongs to the current interval $[\epsilon_i, \epsilon_j]$, and that (P^+, Q^+) is an efficient solution in $\mathcal{S}^+$. When the interval length becomes smaller than 1, we have $\epsilon_i = \epsilon_j = \epsilon^+$. The algorithm then returns (P^+, Q^+), which is the unique efficient solution minimizing P in $\mathcal{S}^+$.

Regarding the number of iterations, the initial length of the interval is $Q_s^- - Q_s^+ \leq Q_{\max}$. At each iteration, the interval length is at most halved. Therefore, after k iterations the length is bounded by $\lceil (Q_s^- - Q_s^+)/2^k\rceil$. The loop terminates when this value is less than 1, which requires

$$k \geq \left\lceil \log_2\left((Q_s^- - Q_s^+) + 1\right)\right\rceil \leq \lceil \log_2 Q_{\max}\rceil + 1.$$

Consequently, SEARCH_Phase_Two_$\mathcal{S}^+$ converges in $\mathcal{O}(\log_2 Q_{\max})$ iterations. □

A key advantage of our two-phase method is that it reduces the task of computing the KS solution to solving two simpler scalarized problems. When both of these can be solved efficiently, the KS solution can be obtained in weakly polynomial time.

Hypothesis 1 $\mathcal{F}(\alpha)$ and $\mathcal{P}(\epsilon)$ can be performed in polynomial time.

Following Theorems 5 and 6, we obtain the theorem below.

Theorem 7 *Consider a BOCO problem whose corresponding weighted sum problem $\mathcal{F}(\alpha)$ and ϵ-constraint problem $\mathcal{P}(\epsilon)$ satisfy Hypothesis 1. Then, the KS solution of this BOCO problem can be determined in weakly polynomial time.*

4 Computational study

This section presents a computational study of the KS solution on two classical and representative BOCO problems. The first is the Bi-Objective Knapsack Problem, which is formulated as a maximization problem; the second is the Bi-Objective Spanning Tree Problem, formulated as a minimization problem. Both problems are NP-hard in the multiobjective setting, in the sense that the number of efficient solutions may grow expotentially with respect to the input size [8]. These two problems are selected to reflect the two most common types of BOCO formulations and to demonstrate the generality of the proposed two-phase approach. For each problem, we describe the mathematical model and implement the two-phase approach to determine the KS solutions. We present numerical results together with clear explanations and observations. Furthermore, we compare the KS solution with other fairness-based selection methods: Proportional Fairness [32] and Max-min for Bi-objective Knapsack Problem, and Nash Fairness [31] and Minmax for Bi-objective Spanning Tree Problem. All computational experiments were conducted on a system equipped with an Intel Core i5-10500 CPU operating at 3.10 GHz, featuring 15 GB of RAM and utilizing 6 cores and 12 threads.

4.1 Bi-Objective Knapsack Problem

The Bi-Objective Knapsack Problem (BOKP) extends the classical 0-1 Knapsack Problem by treating two conflicting objectives, typically maximizing profit while minimizing cost or risk, simultaneously. It appears in many practical settings, such as selecting projects in capital budgeting [42], planning transportation investments [48], and designing environmental cleanup strategies [14].

Several exact methods have been developed to generate the complete Pareto front for the BOKP. A purely theoretical dynamic-programming framework was outlined in [18] without accompanying experiments. Visée et al. [50] introduced the two-phase method, which uses scalarization together with branch-and-bound to recover both supported and non-supported efficient solutions. Captivo et al. [4] transformed the BOKP into a bicriteria shortest-path problem and solved it via a labeling algorithm, while Da Silva et al. [6] proposed core-problem reductions to speed up exact enumeration. In addition, the ϵ-constraint approach, when implemented with an exact solver such as CPLEX, has proven capable of handling larger instances effectively [3]. A detailed overview of exact and heuristic methods for the Multiobjective Multidimensional Knapsack Problem, including its bi-objective variant, is presented in [21].

Our goal in this section is to find the KS solution(s) for the BOKP using the two-phase method and the ϵ-constraint method, as outlined in Section 3. Mathematically, the problem is defined as follows. Given a set of n items, each item i is characterized by two objective values, p_i and q_i (e.g., two different profit measures or utility functions), a weight w_i, and a binary decision variable x_i, where $x_i = 1$ if the item is selected and $x_i = 0$ otherwise. Let $\mathcal{X}$ denote the set of all feasible variables x. As established in Section 2, all profits, weights, and capacities are assumed to be positive integers. This ensures that the objective values of BOKP are also positive whole numbers. The knapsack has a limited capacity C, restricting the total weight of selected items. The objective of BOKP is to maximize both objectives while satisfying the capacity constraint. Formally, the objectives can be written as

$$f_1(x) = \sum_{i=1}^{n} p_i x_i, \quad f_2(x) = \sum_{i=1}^{n} q_i x_i.$$

In this work, we address the problem of finding the KS solution(s) for the BOKP, which we refer to as KS-BOKP. An important note is that the scalarization process in Section 2.2 only affects the values of the objective functions while the solution(s) associated with the KS solution(s) remain unchanged. To solve $\mathcal{F}(\alpha)$ in the first phase of KS-BOKP, we reformulate it as the following single-objective problem

$$\begin{aligned}
\text{(BOKP-I)} \quad \max \quad & \sum_{i=1}^{n} ((1-\alpha)p_i + \alpha q_i)x_i, \quad \alpha \in [0,1] \\
\text{subject to} \quad & \sum_{i=1}^{n} w_i x_i \le C, \quad x_i \in \{0,1\}, \quad \forall i \in \{1,\dots,n\}.
\end{aligned}$$

In the second phase, the problem $\mathcal{P}(\epsilon)$ is similarly reduced to the following form

$$\begin{aligned}
\text{(BOKP-II)} \quad \max \quad & \sum_{i=1}^{n} (p_i + \frac{q_i}{\sum_{i=1}^{n} q_i(x_i^2 - x_i^1)})x_i \\
\text{subject to} \quad & \sum_{i=1}^{n} q_i x_i \ge \epsilon + \sum_{i=1}^{n} q_i x_i^2 \\
& \sum_{i=1}^{n} w_i x_i \le C, \quad x_i \in \{0,1\}, \quad \forall i \in \{1,\dots,n\},
\end{aligned}$$

where $x^1 = \text{argmax}_{x \in \mathcal{X}} \sum_{i=1}^n p_i x_i$, $x^2 = \text{argmax}_{x \in \mathcal{X}} \sum_{i=1}^n q_i x_i$ and $\epsilon > 0$ is given. Observe that the problem BOKP-I is a single-objective knapsack problem with modified profits for each item $\tilde{p}_i(\alpha) = (1-\alpha)p_i + \alpha q_i$, while the problem BOKP-II is the knapsack problem with a side constraint. Some exact methods, such as dynamic programming and branch-and-bound algorithms, have been widely applied to solve classical knapsack problems. These techniques often perform well on small-sized instances, where the number of variables and constraints remains limited. However, these methods do not scale well when the problem size increases or when more constraints are added. Dynamic programming quickly becomes impractical because it needs too much memory and time. Branch-and-bound also struggles as the number of possible solutions grows rapidly. To overcome this issue, we use IBM ILOG CPLEX, a solver that implements the branch-and-cut method to solve mixed-integer linear programming formulations. This approach is particularly effective for solving both BOKP-I and BOKP-II on the instance sizes considered in our experiments, as described in the next subsection.

To show the effectiveness and scalability of the two-phase method in identifying the KS solution for the BOKP, we conducted a series of computational experiments across a wide range of problem instances. These experiments were designed to assess the algorithm's performance under varying problem sizes for a representative capacity tightness level. We generated problem instances with the number of items n ranging from 50 to 1000, increasing by 50 each time. For each instance, item weights w_i were randomly chosen from the range $[1, 20]$. The two profit values, p_i and q_i, were randomly chosen from $[1, 30]$ and $[10, 40]$, resp. To evaluate the algorithm's performance under varying capacity levels, we used the following ratio to set the knapsack capacity C

$$r = \frac{C}{\sum_{i=1}^n w_i}.$$

In this analysis, we focused only on the case where $r = 0.5$. When $r = 0.5$, there are many possible combinations of items to choose from, but the capacity is still limited enough to create strong competition between solutions. This value is known to be the hardest for both single-objective and multi-objective knapsack problems [50]. To ensure reproducibility, each value of n corresponds to a single randomly generated instance constructed exactly according to the distributions described above. Since the KS procedure is deterministic for any fixed instance, the computational results reported for each n contain no stochastic variance. This design enables a clean assessment of scalability with respect to problem size.

The numerical results for the KS-BOKP are summarized in Table 1, which compares the outcomes of Phase I and Phase II computations. The column "n_{items}" indicates the number of items considered, while the columns ρ_{I}^- and ρ_{I}^+ correspond to the KS measures associated with the two consecutive supported efficient solutions located on opposite sides of the Utopia-Nadir line in Phase I. Similarly, ρ_{II}^- and ρ_{II}^+ denote the values of the KS measure computed in Phase II. Following Theorem 1, at least one of the solutions evaluated in Phase II represents the KS solution. The KS solutions are highlighted in bold within the corresponding computations. The columns

Table 1: Computational results for Phase I and Phase II of KS-BOKP

n_{items}	ρ_{I}^{-}	ρ_{I}^{+}	Time_{I} (s)	Iter_{I}	ρ_{II}^{-}	ρ_{II}^{+}	Time_{II} (s)	Iter_{II}
50	0.1219	0.2263	1.7841	16	0.0099	0.0389	3.6863	8
100	0.0668	0.1305	1.7741	19	0.0100	0.0062	4.3625	9
150	0.0974	0.0399	1.3548	20	0.0148	0.0010	3.3736	9
200	0.0343	0.0381	1.9114	20	0.0034	0.0013	4.4721	7
250	0.0221	0.0854	2.1135	21	0.0017	0.0014	5.9898	10
300	0.0530	0.0097	2.3678	22	0.0011	0.0022	6.1874	9
350	0.0120	0.0217	2.2881	22	0.0023	0.0001	7.5298	8
400	0.0349	0.0296	2.7803	22	0.0016	0.0006	8.4721	11
450	0.0238	0.0279	2.9490	23	0.0020	0.0004	7.3836	10
500	0.0114	0.0044	2.7021	23	0.0010	0.0008	6.1546	6
550	0.0056	0.0708	3.1520	24	0.0006	0.0007	8.1656	10
600	0.0145	0.0080	2.8486	24	0.0001	0.0017	6.8889	8
650	0.0142	0.0342	3.1322	24	0.0002	0.0022	7.7973	11
700	0.0117	0.0281	3.9626	24	0.0011	0.0000	7.3023	10
750	0.0121	0.0090	3.9525	24	0.0006	0.0005	7.0461	8
800	0.0088	0.0058	4.0103	25	0.0011	0.0002	8.7449	8
850	0.0025	0.0144	4.4954	25	0.0009	0.0001	9.2164	8
900	0.0238	0.0042	4.8531	25	0.0001	0.0008	9.8973	10
950	0.0359	0.0039	4.6273	25	0.0007	0.0007	9.9554	11
1000	0.0253	0.0109	4.8708	25	0.0005	0.0006	11.0097	11

"Time (s)" and "Iters" report the computational time (in seconds) and the number of iterations required to obtain the solutions in each phase, resp.

The results show that, even though it is theoretically possible to have two KS solutions, every computational experiment produced only one KS solution for each instance. Additionally, while the KS solution could, in theory, be a supported efficient solution, all the KS solutions found were non-supported. This point is aligned with theoretical predictions and suggests a clear trend: KS solutions are usually non-supported. When examining the values of the KS measure ρ, it becomes clear that the solutions from Phase II provide a much better trade-off between the two objectives than those from Phase I. In particular, the average ρ value in Phase I is about 0.0871, while in Phase II, it decreases to 0.0185. This corresponds to a relative reduction of approximately 78.75% in the KS measure. Also, in all instances, none of the KS solutions lies exactly on the Utopia-Nadir line, as evidenced by the fact that a KS measure equal to zero was not observed. In terms of computational cost, both phases require a relatively small number of iterations. In Phase I, the number of iterations increases slightly with the number of items, but this growth remains modest thanks to the logarithmic behavior of the underlying algorithm with respect to $P_{\max}$ and $Q_{\max}$. Interestingly, although the BOKP solved in Phase II is theoretically more difficult, the time needed to compute the KS solution does not always grow with problem size. For instance, when $n = 400$, Phase II required about 8.47 seconds, whereas for $n = 500$ it took only about 6.15 seconds. This variation can be explained by the number of iterations: 11 in the former case, compared to just 6 in the latter. More precisely, the number of iterations in Phase II does not scale predictably with problem size, since its performance depends more on the solutions provided by Phase I and, implicitly, on the structure of the problem's feasible set.

Overall, the empirical behaviour of the two-phase method is consistent with its theoretical guarantees. Phase I exhibits a mild increase in running time as n grows, reflecting the logarithmic bound in Theorem 5, since $P_{\max}$ and $Q_{\max}$ scale linearly with n. Phase II shows non-monotonic running times because the number of iterations depends on the width of the interval $[Q_s^+, Q_s^-]$ delivered by Phase I rather than on n. In several instances, larger problems even require fewer iterations, demonstrating that the two-phase method remains computationally stable and scalable up to $n = 1000$.

Moreover, to deepen the comparative analysis, we evaluate the KS solutions obtained in the previous experiments against two alternative fairness-based approaches: the Proportional Fairness (PF) solution [32] and the Maxmin solutions for the BOKP. All three methods aim to identify fair trade-offs between the two conflicting objectives f_1 and f_2, but they are grounded in fundamentally different fairness criteria.

The *PF solution*, introduced in [32], is based on the principle of proportional fairness. A feasible solution $(f_1^{\mathrm{PF}}(x), f_2^{\mathrm{PF}}(x))$ is considered PF solution if it satisfies the following inequality for all other feasible solutions $(f_1(x), f_2(x))$

$$\frac{f_1(x) - f_1^{\mathrm{PF}}(x)}{f_1^{\mathrm{PF}}(x)} + \frac{f_2(x) - f_2^{\mathrm{PF}}(x)}{f_2^{\mathrm{PF}}(x)} \leq 0.$$

This condition ensures that no other solution provides simultaneous proportional improvements in both objectives. Importantly, the PF solution, when it exists, also coincides with the solution that maximizes the product

$$f_1(x) \cdot f_2(x),$$

which provides an intuitive balance between the two objectives. However, due to the discrete nature of the solution space in combinatorial optimization, the PF solution does not always exist. In our experiments, among 20 instances of the BOKP, PF solutions exist for only 15 instances, meaning that there is no PF solution in 20% of the instances.

On the other hand, the *Maxmin solution* is based on the max-min fairness principle. It selects the solution that maximizes the minimum of the two objectives

$$\max_{x \in \mathcal{X}} \{\min(f_1(x), f_2(x))\}.$$

This criterion emphasizes protecting the worst-performing objective, thereby ensuring a conservative yet balanced compromise. The Maxmin solution always exists in finite discrete settings and is particularly relevant when ensuring the minimal acceptable performance for both objectives is critical.

To enable a consistent comparison between the three fairness-based criteria (PF, Maxmin, and KS) in the context of the BOKP, which is formulated as a maximization problem, we adopt their maximization-equivalent representations. While the KS solution was originally defined in Section 2.2 under a bi-objective minimization framework, the KS rule naturally extends to maximization through the standard sign-reversal transformation $f_i \mapsto -f_i$. Owing to the affine invariance of the KS solution, this transformation preserves all theoretical properties while allowing the KS measure to be expressed in a form compatible with maximization-based criteria.

Let x^1 and x^2 denote the solutions that maximize f_1 and f_2, resp. For a bi-objective maximization problem, the KS solution can be obtained by solving

$$\max_{x \in \mathcal{X}} \min \left(\frac{f_1(x) - f_1(x^2)}{f_1(x^1) - f_1(x^2)}, \frac{f_2(x) - f_2(x^1)}{f_2(x^2) - f_2(x^1)} \right),$$

where $\mathcal{S}$ denotes the Pareto front. This expression is algebraically equivalent to the minimization-based KS measure introduced earlier, but reformulated such that larger values correspond to more balanced outcomes, thereby aligning with the maximization orientation of PF and Maxmin. Importantly, this reformulation does not alter the KS solution itself; it merely provides a unified evaluation framework for the purpose of comparison.

Accordingly, the three fairness criteria adopted in this study, each in maximization form, are expressed by the following formulations

$$\begin{aligned}
\text{KS:} \quad & \min \left(\frac{f_1(x^1) - f_1(x)}{f_1(x^1) - f_1(x^2)}, \frac{f_2(x^2) - f_2(x)}{f_2(x^2) - f_2(x^1)} \right), \\
\text{PF:} \quad & f_1(x) \cdot f_2(x), \\
\text{Maxmin:} \quad & \min(f_1(x), f_2(x)).
\end{aligned}$$

We report in Table 2 the average percentage gap of each solution method when evaluated using fairness criteria different from the one it optimizes. The results are computed over the 20 BOKP instances described above. Each value indicates the average relative performance loss (in %) when replacing a reference solution by another, under a given criterion. For instance, the value "-4.48%" in the first row and second column means that using the Maxmin solution instead of the PF solution results in a 4.48% decrease in performance when evaluated by the PF criterion. By definition, values on the diagonal are zero, indicating that each solution performs best under its criterion.

Table 2: Comparison of fairness criteria for the BOKP (20 instances)

Solution / Criterion	PF	Maxmin	KS
PF	0.00	-4.48	-4.55
Maxmin	-8.38	0.00	-101.18
KS	-0.23	-4.99	0.00

The results show that each solution performs best under its criterion, as expected. However, some methods also perform competitively across other evaluation measures.

In particular, the KS solution is the most robust: it has the smallest average gap when evaluated by PF and performs reasonably well on Maxmin. On the other hand, the Maxmin solution suffers a severe performance drop when assessed by the KS criterion, indicating that it may not be well-balanced with respect to the shifted trade-off between objectives. In general, this comparison emphasizes the importance of selecting the appropriate evaluation criterion based on the desired balance or fairness between objectives.

4.2 Bi-Objective Spanning Tree Problem

The Bi-Objective Spanning Tree Problem (BOSTP) is an important extension of the classical Minimum Spanning Tree problem, which seeks to construct a spanning tree that simultaneously minimizes two objective functions. This problem arises in practical applications such as network design, where objectives like cost minimization and reliability maximization are often in conflict [39], and transportation systems, where trade-offs between efficiency and environmental impact must be considered [17]. In this section, we focus on a variant of the BOSTP in which the two objectives are the total cost and the total time of the spanning tree. The problem of computing the Pareto front for BOSTP under these two criteria has been widely recognized as important and has been addressed by several studies in the literature [40, 46, 45, 43]. For simplicity of computation, we assume that both cost and time values are positive integers. As a result, all objective values in the BOSTP are also positive integers, as stated in Section 2.1.

Formally, let $G = (V, E)$ be a connected undirected graph with $|V| = n$ nodes and $|E| = m$ edges. Each edge $e \in E$ is assigned two strictly positive weights: c_e and t_e, representing cost values and time values, resp. A feasible solution is any spanning tree $T \subseteq E$, i.e., a subset of $n-1$ edges that connects all nodes without any cycles. Let $x = (x_e)_{e\in E} \in \{0,1\}^m$ be a binary decision vector corresponding to the spanning tree T, where $x_e = 1$ indicates that edge e is selected in T, the two objective functions are defined as

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

Let $\mathcal{T}$ be the set of all feasible solutions, i.e., all spanning trees of G. In this work, we aim to determine the KS solution for BOSTP using the two-phase procedure described in Section 3. We refer to this problem as KS-BOSTP. As mentioned in the previous section, an important feature of the scalarization method is that it changes only the objective values, while the constraints stay the same. Based on this, the first phase involves solving the following single-objective problem

$$\text{(BOSTP-I)} \quad \min \quad \sum_{e\in E} ((1-\alpha)c_e + \alpha t_e)x_e, \quad \alpha \in [0,1] \tag{7a}$$

$$\text{subject to} \quad \sum_{e\in E} x_e = |V| - 1 \tag{7b}$$

$$\sum_{e\in\delta(V')} x_e \geq 1,\ \forall V' \subset V,\ 2 \leq |V'| \leq |V|-2 \tag{7c}$$

$$x_e \in \{0,1\} \quad \forall e \in E \tag{7d}$$

In this formulation, constraint (7b) ensures that the selected edges form a tree with exactly $n-1$ edges. Constraint (7c), known as the *subtour elimination constraint*, guarantees the connectivity of the selected edges. It requires that for every non-empty proper subset $V' \subset V$, there must be at least one edge crossing between V and $V \setminus V'$.

In the second phase, we present the following formulation of problem $\mathcal{P}(\epsilon)$, which includes all constraints from (7b) to (7d). To avoid redundancy, these constraints are not explicitly restated here. Problem $\mathcal{P}(\epsilon)$ can thus be equivalently reformulated as follows

$$\text{(BOSTP-II)} \quad \min \quad \sum_{e\in E}(c_e + \frac{t_e}{\sum_{e\in E} t_e(x_e^1 - x_e^2)})x_e \tag{8a}$$

$$\text{subject to} \quad \sum_{e\in E} t_e x_e \leq \epsilon + \sum_{e\in E} t_e x_e^2 \tag{8b}$$

where $x_1^* = \text{argmin}_{x\in\mathcal{T}} \sum_{e\in E} c_e x_e$, $x_2^* = \text{argmin}_{x\in\mathcal{T}} \sum_{e\in E} t_e x_e$ and $\epsilon > 0$ is given. Observe that the problem BOSTP-I reduces to a standard single-objective minimum spanning tree problem, where each edge is assigned a scalarized weight $w_e = (1-\alpha)c_e + \alpha t_e$. As such, BOSTP-I can be solved in polynomial time. In contrast, BOSTP-II corresponds to a spanning tree problem with an additional side constraint. This class of problems is widely known as the *Constrained Minimum Spanning Tree Problem* (CMSTP) [41], which is weakly NP-hard [1].

To solve the single-objective minimum spanning tree problem BOSTP-I, we use the default algorithm implemented in NetworkX, which relies on Kruskal's algorithm [19]. For the BOSTP-II problem, the additional constraint involving the total time makes it impossible to use classical MST algorithms. A natural approach is to use the classical cut-based model and add violated connectivity constraints lazily during branch-and-bound. However, this method often slows down the solver and leads to unstable performance when combined with side constraints. Instead, we use the single-commodity flow (SCF) [22] formulation to model connectivity. The SCF model has a polynomial number of constraints and is more stable in practice, especially for small to medium graphs [22]. Following the experimental design described in [33], we conducted experiments on a set of randomly generated graphs using the NetworkX library. The graph instances were generated based on the Erdos-Renyi $G_{n,p}$ model [9], where n represents the number of nodes and p is the probability of edge creation. In our experiments, the number of nodes n varied from 20 to 300 in increments of 20. For each n, three graph instances were generated: Ins$n$_1, with $p = 0.3$, Ins$n$_2, with $p = 0.5$, Ins$n$_3, with $p = 0.7$. Each edge in the graphs was assigned a random cost and a random time, independently drawn from uniform distributions $[20, 120]$ and $[1, 100]$, resp.

The numerical results for the KS-BOSTP are presented in Table 3, which compares the outcomes of Phase I and Phase II. Each row shows the results for one instance. The

Table 3: Computational results for Phase I and Phase II of KS-BOSTP

Instance	ρ_{I}^{-}	ρ_{I}^{+}	$\mathrm{Time}_{\mathrm{I}}$ (s)	$\mathrm{Iters}_{\mathrm{I}}$	ρ_{II}^{-}	ρ_{II}^{+}	$\mathrm{Time}_{\mathrm{II}}$ (s)	$\mathrm{Iters}_{\mathrm{II}}$
Ins20_1	0.0330	0.0150	0.0128	21	0.0078	0.0150	0.9465	8
Ins20_2	0.0802	0.0725	0.0156	21	0.0008	0.0142	0.4034	10
Ins20_3	0.0246	0.0677	0.0214	22	0.0003	0.0072	0.7223	10
Ins40_1	0.0384	0.0019	0.0423	24	0.0039	0.0019	1.4729	10
Ins40_2	0.0006	0.0110	0.0619	24	0.0006	0.0030	1.2326	6
Ins40_3	0.0096	0.0040	0.0814	24	0.0018	0.0015	1.5764	8
Ins60_1	0.0099	0.0055	0.0895	25	0.0012	0.0009	2.3126	8
Ins60_2	0.0034	0.0007	0.1355	25	0.0010	0.0007	0.9133	6
Ins60_3	0.0168	0.0070	0.1962	26	0.0008	0.0012	6.9012	10
Ins80_1	0.0072	0.0054	0.1527	26	0.0003	0.0006	18.6176	10
Ins80_2	0.0054	0.0053	0.2389	26	0.0012	0.0001	15.6491	8
Ins80_3	0.0082	0.0085	0.6761	26	0.0002	0.0011	19.3435	8
Ins100_1	0.0055	0.0056	0.3473	27	0.0006	0.0012	31.1193	7
Ins100_2	0.0094	0.0052	0.3921	27	0.0001	0.0010	22.7197	10
Ins100_3	0.0064	0.0017	0.5316	26	0.0006	0.0005	12.6105	7
Ins120_1	0.0025	0.0048	0.3410	27	0.0004	0.0013	25.1212	10
Ins120_2	0.0046	0.0019	0.5660	27	0.0002	0.0012	20.5397	7
Ins120_3	0.0016	0.0034	0.8578	27	0.0001	0.0005	46.4269	8
Ins140_1	0.0015	0.0086	0.5138	28	0.0008	0.0013	26.9239	10
Ins140_2	0.0033	0.0008	0.8108	28	0.00004	0.0003	43.3445	8
Ins140_3	0.0015	0.0025	1.0834	27	0.0003	0.0004	47.4152	8
Ins160_1	0.0015	0.0060	0.9113	28	0.0003	0.0022	158.1022	10
Ins160_2	0.0094	0.0071	1.0361	28	0.0002	0.0011	276.6988	12
Ins160_3	0.0023	0.0031	1.4636	28	0.0001	0.0011	96.5159	7
Ins180_1	0.0098	0.0020	1.8227	29	0.0001	0.0010	236.3378	12
Ins180_2	0.0011	0.0043	1.3655	28	0.0001	0.0013	126.1829	7
Ins180_3	0.0058	0.0020	1.9526	28	0.0004	0.0009	324.5765	10

values ρ_{I}^{-} and ρ_{I}^{+} represent the KS measures obtained from two supported efficient solutions found in Phase I, located on opposite sides of the Utopia-Nadir line. Similarly, ρ_{II}^{-} and ρ_{II}^{+} are the values of the KS measure computed in Phase II. According to Theorem 1, at least one of the two solutions from Phase II is the KS solution. The columns "$\mathrm{Time}_{\mathrm{I}}$" and "$\mathrm{Time}_{\mathrm{II}}$" indicate the time in seconds to solve each phase, while "$\mathrm{Iters}_{\mathrm{I}}$" and "$\mathrm{Iters}_{\mathrm{II}}$" show the number of iterations used.

Our results show that for every tested instance of the BOSTP, the KS solution is always unique and is found only in Phase II. This confirms that Phase I alone is not sufficient to reach the KS solution. Moreover, none of the KS solutions is supported. While it is theoretically possible for a KS solution to lie on the supported part of the Pareto front, all the solutions found in our experiments are non-supported. Examining the values of the KS measure ρ shows that Phase II consistently yields a better balance between the two objectives than Phase I. On average, the value of the KS measure ρ in Phase I is approximately 0.0197, while in Phase II it drops to around 0.0013. This corresponds to a relative reduction of about 93.4% in the KS measure, highlighting the effectiveness of Phase II in improving the balance between cost and time.

In terms of computation, Phase I is very fast and stable. It usually finishes in less than 0.6 seconds and needs between 21 and 29 iterations. This good performance is due to its binary search design. Phase II takes more time, especially on larger graphs. For example, Ins160_2 and Ins180_3 required more than 200 seconds. However, the time and number of iterations in Phase II do not grow monotonically with graph size. Instead, they depend more on the geometry of the feasible efficient set induced by Phase I. This indicates that the difficulty of Phase II is driven more by problem structure than by instance size alone.

To better understand how fairness can be interpreted in the context of BOSTP, which involves the simultaneous minimization of two objectives, we compare the KS solution with two alternative fairness-based solutions tailored to minimization: the Nash Fairness (NF) solution [31] and the Minmax solution.

The *NF solution*, introduced in [31], is based on the principle of Nash fairness. A feasible solution x^{NF} satisfies this condition if, for all other feasible solutions $x \in \mathcal{X}$,

$$\frac{f_1(x) - f_1(x^{\mathrm{NF}})}{f_1(x^{\mathrm{NF}})} + \frac{f_2(x) - f_2(x^{\mathrm{NF}})}{f_2(x^{\mathrm{NF}})} \geq 0.$$

This condition ensures that no other solution can reduce both objectives proportionally at the same time. Intuitively, it expresses resistance to joint improvement and leads to a balanced trade-off where reducing one objective further would require sacrificing the other. Mathematically, this condition characterizes x^{NF} as a local minimizer of the function $\log(f_1(x)) + \log(f_2(x))$, assuming $f_1(x), f_2(x) > 0$. Since the logarithm is strictly increasing, this is equivalent to minimizing the product $f_1(x) \cdot f_2(x)$. This multiplicative criterion reflects a natural compromise: any deviation from the NF solution would increase the overall "combined cost" of the two objectives in a geometric sense. In our experiments, NF solutions were found in all instances of the BOSTP and were non-unique in 50% of the instances.

The *Minmax solution*, in another principle, focuses on controlling the worst-case performance across the two objectives. It selects the solution that minimizes the maximum of $f_1(x)$ and $f_2(x)$, ensuring that no single objective dominates the trade-off too severely. To summarize, the three fairness criteria, in minimization form, used in this study are

$$\begin{aligned} \text{KS:} &\quad \max\left(\frac{f_1(x) - f_1(x_1^*)}{f_1(x_2^*) - f_1(x_1^*)}, \frac{f_2(x) - f_2(x_2^*)}{f_2(x_1^*) - f_2(x_2^*)}\right), \\ \text{NF:} &\quad f_1(x) \cdot f_2(x), \\ \text{Minmax:} &\quad \max(f_1(x), f_2(x)). \end{aligned}$$

Table 4: Comparison of fairness criteria for the BOSTP (27 instances)

Solution / Criterion	NF	Minmax	KS
NF	0.00	85.58	337.37
Minmax	102.50	0.00	110.02
KS	63.81	18.44	0.00

Table 4 reports the average percentage gap when each solution is evaluated using fairness criteria different from the one it optimizes. Each entry indicates the average relative performance loss (in%) computed over the full set of 27 BOSTP instances described above. While the KS and NF solutions are available for all 27 instances, the Minmax solution could only be computed for 8 instances due to time limits reached on the remaining instances, all involving graphs with at least 80 nodes.

As shown in Table 4, the three solutions display different levels of compatibility when evaluated under fairness criteria other than their own. The NF and Minmax solutions exhibit high sensitivity to the choice of evaluation criterion, as reflected by their large average deviations. In particular, each performs well only under the criterion it optimizes, but incurs significant losses under others. By comparison, the KS solution achieves more consistent performance across all criteria, with the lowest average deviation when evaluated under alternative fairness views. This suggests that the KS solution provides a more balanced compromise in situations where the definition of fairness is not predetermined or may vary.

5 Conclusion

This paper investigated the Kalai-Smorodinsky (KS) solution in the context of bi-objective combinatorial optimization (BOCO). Since the classical continuous definition is not directly applicable in discrete settings, we introduced a discrete variant defined over the Pareto-optimal set and established that a KS solution always exists, with at most two such solutions. Building on this characterization, we developed a two-phase procedure that combines weighted sum scalarization with a refined ϵ-constraint search. The method exhibits logarithmic complexity and provides a practical way to compute KS solutions in general BOCO problems. The approach was applied to two representative NP-hard problems, the Bi-objective Knapsack Problem and the Bi-objective Spanning Tree Problem. The computational study shows that the KS solution is typically non-supported and is consistently identified in the second phase. Moreover, a comparative analysis with Nash Fairness, Proportional Fairness, and Minmax solutions highlights the robustness of the KS solution under different fairness criteria.

This work opens several directions for further research. A key challenge lies in Phase II of our method, which targets non-supported efficient solutions that cannot be found through weighted sum scalarization. As identifying such solutions is computationally demanding, improving the second phase through approximation algorithms would be highly beneficial. In parallel, we aim to extend the KS solution beyond the bi-objective setting. A natural extension of this work is to study the three-objective case, including its formal definition, structural characteristics, and algorithmic aspects. Further, we aim to investigate how the approach can be generalized to multi-objective problems.

References

[1] V. Aggarwal, Y. Aneja, and K. Nair. Minimal spanning tree subject to a side constraint. *Computers and Operations Research*, 9:287–296, 1982.

[2] L. Amorosi and J. Puerto. Two-phase strategies for the bi-objective minimum spanning tree problem. *International Transactions in Operational Research*, 29:3435–3463, 2022.

[3] C. Bazgan, M. Hugot, and D. Vanderpooten. Solving the 0-1 multi-objective knapsack problem with very large instances. *European Journal of Operational Research*, 200(3):847–854, 2010.

[4] M. E. Captivo, J. C. N. Clímaco, J. R. Figueira, E. Q. V. Martins, and J. L. Santos. Solving bicriteria 0-1 knapsack problems using a labeling algorithm. *Computers and Operations Research*, 30(12):1865–1886, 2003.

[5] Y. Collette and P. Siarry. *Multiobjective Optimization: Principles and Case Studies*. Springer, 2003.

[6] C. G. Da Silva, J. C. N. Clímaco, and J. R. Figueira. Core problems in the bi-criteria 0-1 knapsack: new developments. Technical Report 12, INESC-Coimbra, 2005.

[7] K. Deb and K. Miettinen. Nadir point estimation using evolutionary approaches: Better accuracy and computational speed through focused search. In *Lecture Notes in Economics and Mathematical Systems*, volume 634, pages 339–354. Springer, 2010.

[8] M. Ehrgott. *Multicriteria Optimization*. Springer, 2005.

[9] P. Erdos and A. Renyi. On random graphs. *Publ. Math. Debrecen*, 6:290–297, 1959.

[10] C. Filippi and E. Stevanato. A two-phase method for bi-objective combinatorial optimization and its application to the TSP with profits. *Algorithmic Operations Research*, 7:125–139, 2013.

[11] Y. Y. Haimes, L. S. Lasdon, and D. A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, 1(3):296–297, 1971.

[12] C.-L. Hwang and K. Yoon. *Multiple Attribute Decision Making: Methods and Applications*. Springer-Verlag, 1981.

[13] L. Iorio, L. Fourment, S. Marie, and M. Strano. Application of the Kalai-Smorodinsky approach in multi-objective optimization of metal forming processes. *International Journal of Material Forming*, 10:515–526, 2017.

[14] L. Jenkins. A bicriteria knapsack program for planning remediation of contaminated lightstation sites. *European Journal of Operational Research*, 140(2):427–433, 2002.

[15] E. Kalai and M. Smorodinsky. Other solutions to Nash's bargaining problem. *Econometrica*, 43(3):513–518, 1975.

[16] D. Karos, N. Muto, and S. Rachmilevitch. A generalization of the Egalitarian and the Kalai-Smorodinsky bargaining solutions. *International Journal of Game Theory*, 47(4):1169–1182, 2018.

[17] K. Klamroth. *Single and Multiple Objective Minimum Spanning Tree Problems: Mathematical Programming Models and Solution Procedures*. PhD thesis,

University of Kaiserslautern, 2002.

[18] K. Klamroth and M. Wiecek. Dynamic programming approaches to the multiple criteria knapsack problem. *Naval Research Logistics*, 47(1):57–76, 2000.

[19] J. B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7:48–50, 1956.

[20] H. Lee and P. S. Pulat. Bicriteria network flow problems: integer case. *European Journal of Operational Research*, 66:148–157, 1993.

[21] T. Lust and J. Teghem. The multiobjective multidimensional knapsack problem: a survey and a new approach. *International Transactions in Operational Research*, 19(4):495–520, 2012.

[22] T. L. Magnanti and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

[23] M. Mariotti. Nash bargaining theory when the number of alternatives can be finite. *Social Choice and Welfare*, 15:413–421, 1998.

[24] A. Mas-Colell, M. Whinston, and J. Green. *Microeconomic Theory*. Oxford University Press, 1995.

[25] G. Mavrotas. Effective implementation of the epsilon-constraint method in multiobjective mathematical programming problems. *Applied Mathematics and Computation*, 213:455–465, 2009.

[26] G. Mavrotas and D. Diakoulaki. A branch and bound algorithm for mixed 0-1 multiple objective linear programming. *European Journal of Operational Research*, 107(3):530–541, 1998.

[27] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic, Boston, 1999.

[28] H. Moulin. Implementing the kalai-smorodinsky bargaining solution. *Journal of Economic Theory*, 33(1):32–45, 1984.

[29] 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.

[30] J. F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.

[31] M. H. Nguyen, M. Baiou, V. H. Nguyen, and T. Q. T. Vo. Generalized nash fairness solutions for bi-objective minimization problems. *Networks*, 83:83–99, 2023.

[32] M. H. Nguyen, M. Baïou, V. H. Nguyen, and T. Q. T. Vo. Proportional fairness for combinatorial optimization. *Theoretical Informatics, Lecture Notes in Computer Science (LATIN 2024)*, 14579, 2024.

[33] T. L. Nguyen, V. H. N. V. Hung, M. H. Nguyen, and T. V. T. Vu. On the kalai-smorodinsky solutions for bi-objective spanning tree problem. In *Algorithmic Aspects in Information and Management*, pages 86–98. Springer Nature Singapore, 2024.

[34] S. Opricovic and G.-H. Tzeng. Compromise solution by MCDM methods: A comparative analysis of VIKOR and TOPSIS. *European Journal of Operational Research*, 156(2):445–455, 2004.

[35] O. Özpeynirci and M. Köksalan. An exact algorithm for finding extreme supported non-dominated points of multiobjective mixed integer problems.

Management Science, 56(12):2302–2315, 2010.

[36] A. Przybylski and X. Gandibleux. Multi-objective branch and bound. *European Journal of Operational Research*, 260(3):856–872, 2017.

[37] A. Przybylski, X. Gandibleux, and M. Ehrgott. Two phase algorithms for the bi-objective assignment problem. *European Journal of Operational Research*, 185:509–533, 2008.

[38] A. Przybylski, X. Gandibleux, and M. Ehrgott. A two phase method for multi-objective integer programming and its application to the assignment problem with three objectives. *Discrete Optimization*, 7(3):149–165, 2010.

[39] A. Raith and G. Ehrgott. A comparison of solution strategies for biobjective shortest path problems. *Computers and Operations Research*, 36(6):1299–1331, 2009.

[40] R. M. Ramos, S. Alonso, J. Sicilia, and C. González. The problem of the optimal biobjective spanning tree. *European Journal of Operational Research*, 111:617–628, 1998.

[41] R. Ravi and M. X. Goemans. The constrained minimum spanning tree problem. In *Proceedings of the Scandinavian Workshop on Algorithm Theory (SWAT)*, volume 1097 of *Lecture Notes in Computer Science*, pages 66–75. Springer, 1996.

[42] M. J. Rosenblatt and Z. Sinuany-Stern. Generating the discrete efficient frontier to the capital budgeting problem. *Operations Research*, 37(3):384–394, 1989.

[43] J. L. Santos, L. D. P. Pugliese, and F. Guerriero. A new approach for the multiobjective minimum spanning tree. *Computers and Operations Research*, 98:69–83, 2018.

[44] A. Sedeño-Noda and C. González-Martín. An algorithm for the biobjective integer minimum cost flow problem. *Computers and Operations Research*, 28(2):139–156, 2001.

[45] F. Sourd and O. Spanjaard. A multiobjective branch-and-bound framework: Application to the biobjective spanning tree problem. *INFORMS Journal on Computing*, 20(3):472–484, 2008.

[46] S. Steiner and T. Radzik. Computing all efficient solutions of the biobjective minimum spanning tree problem. *Computers and Operations Research*, 35:198–211, 2008.

[47] R. E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application*. Wiley, 1985.

[48] J. Teng and G. Tzeng. A multiobjective programming approach for selecting non-independent transportation investment alternatives. *Transportation Research Part B: Methodological*, 30(4):291–307, 1996.

[49] E. L. Ulungu and J. Teghem. The two phases method: An efficient procedure to solve bi-objective combinatorial optimization problems. *Foundations of Computing and Decision Sciences*, 20:149–165, 1995.

[50] M. Visée, J. Teghem, M. Pirlot, and E. L. Ulungu. Two-phases method and branch and bound procedures to solve the bi-objective knapsack problem. *Journal of Global Optimization*, 12:139–165, 1998.

[51] Y. Xu and N. Yoshihara. An equitable nash solution to nonconvex bargaining problems. *International Journal of Game Theory*, 48:769–779, 2019.

[52] R. R. Yager. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1):183–190, 1988.

[53] K. Yoon and C.-L. Hwang. *Multiple Attribute Decision Making: An Introduction.* Sage Publications, 1995.