Yugoslav Journal of Operations Research
33 (2023), Number 2, 309–322
DOI: https://doi.org/10.2298/YJOR220516037N

THE INVERSE k-MAX COMBINATORIAL OPTIMIZATION PROBLEM

Tran Hoai Ngoc NHAN
Faculty of Basic Sciences, Vinh Long University of Technology Education, Vinh Long, Vietnam
tranhoaingocnhan@gmail.com

Kien Trung NGUYEN*
Department of Mathematics, Teacher College, Can Tho University, Can Tho, Vietnam
**Corresponding Author: trungkien@ctu.edu.vn*

Nguyen Thanh HUNG
Department of Mathematics, Teacher College, Can Tho University, Can Tho, Vietnam
nthung@ctu.edu.vn

Nguyen Thanh TOAN
Faculty of Mathematics and Computer Science, University of Science, Ho Chi Minh City, Vietnam
Vietnam National University, Ho Chi Minh City, Vietnam
Faculty of Basic Sciences, Vinh Long University of Technology Education, Vinh Long, Vietnam
toannt@vlute.edu.vn

Received: May 2022 / Accepted: December 2022

Abstract: Classical combinatorial optimization concerns finding a feasible subset of a ground set in order to optimize an objective function. We address in this article the inverse optimization problem with the k-max function. In other words, we attempt to perturb the weights of elements in the ground set at minimum total cost to make a predetermined subset optimal in the fashion of the k-max objective with respect to the perturbed weights. We first show that the problem is in general NP-hard. Regarding the case of independent feasible subsets, a combinatorial $O(n^2 \log n)$ time algorithm is developed, where n is the number of elements in E. Special cases with improved complexity are also discussed.

Keywords: Inverse optimization, k-max, bottleneck, convex, combinatorial algorithms.

MSC: 90-08, 90B99, 90C27.

1. INTRODUCTION

Combinatorial optimization is undoubtedly an interesting topic in operations research due to its potential importance in practical and theoretical contributions. In a combinatorial optimization problem, a ground set and a class of its subsets (the so-called feasible solutions) are given. Then, we aim to find a feasible solution that optimizes an objective function determined by a decision maker. Inverse optimization problem is, on the counterpart, to adjust relative parameters at least in order to make a predetermined feasible solution optimal with respect to the adjusted parameters. The problem, defined in such a way, has become a growing topic with intensively investigations and promising applications. For recent models and solution approaches concerning this topic, one can read the survey of Heuberger [1]. In what follows, let us discuss some existing inverse optimization problems.

The first paper concerning the inverse optimization problem investigated the inverse shortest path problem; see Burton and Toint [2]. They also focused on the application of the problem in earthquake prediction. Then, Ahuja et al. [3] proved that the inverse linear programming problem can be reduced to the problem of the same type by applying the condition of complementary slackness. These pioneering researches set up a stone for later investigation on the inverse optimization. The minimum spanning tree problem is a key topic in combinatorial optimization and its inverse problem was worthwhile to be further studied with efficient solution algorithm; see [4, 5, 6]. Also, [7, 8] developed combinatorial algorithms for the inverse network flows problems under different cost norms based on their special structure. The inverse version of matroid theory, which integrates graph theory and linear algebra, was formulated and solved by Cai and Li [9], Zhang et al. [10]. The inverse $\{0,1\}$-knapsack problem under the rectilinear or Chebyshev norm was modeled as a bilevel integer programming formulation by Roland et al. [11]. He also showed numerical results for the efficiency of the solution approaches. Chung and Demange [12] considered the inverse travelling salesman problem and proposed the approximation result for the problem. One particular case of the general inverse optimization theory, the so-called inverse location problem has been growing with special interest; we can refer to [13, 14, 15, 16, 17, 18] and the concerning references for models, solution algorithms and potential applications.

Modern optimization problem further focused on universal approaches for a wide class of objective functions given by a decision maker. Concerning this purpose, the ordered median function was coined by Nickel and Puerto [19] to generalize a class of functions, where the median and the center functions are among them. Then, the inverse ordered median location problem was also taken into account, for instance, in [20, 21, 22, 23]. The following facts motivates further research on the topic of inverse combinatorial optimization with universal objective function.

1. In optimization theory, a bottleneck problem means to find a feasible solution on the ground set in order to minimize the maximum weight of element in the solution set. On the other hand, the inverse bottleneck problem aims to modify the parameters at minimum cost so that a prespecified feasible solution become the bottleneck of the problem, i.e., its maximum weighted element is the smallest one in comparison to other solutions. The inverse bottleneck problem was intensively studied under various cost functions, we can read [24, 25, 26] for references.

2. By applying the ordered median function, the k-max objective function can be considered as a generalization of the bottleneck problem for $k = 1$. Precisely, the objective focuses on the k^{th} maximum weight of elements in the feasible set instead of the maximum one as in the bottleneck problem. The non-inverse k-max problem can be formulated as a linear integer program by Gorski and Ruzika [27]. Numerical experiments showed the efficiency of the corresponding formulation.

3. The inverse k-max problem is consequently a generalization of the inverse bottleneck problem and plays an important role in exploring the structure of the inverse bottleneck problem as well as in practical situation, where the decision maker tends to choose the k^{th} best of the items in a feasible solution. In spite of its importance, the inverse k-max optimization problem has not been studied so far. For example, in network design the bottleneck value is so vague to measure and it is far from being interesting to the community, then a reasonable presentable value of the network should be given. The k-max is a candidate one as the decision make should focus on the k^{th} maximum weight among the elements in a feasible set. The inverse problem plays the role as resetting the network parameters in an exact way so that the desired feasible solution satisfies the outcome of the decision maker.

In this paper the inverse k-max optimization problem is considered with the ground set and the set of all feasible solutions being the input. This assumption is reasonable. For example, we can exactly know and store the set of spanning trees or the set of paths connecting two vertices in a small-scaled network problem. For the organization of the paper, Section 2 introduces a general setting and optimality condition for the inverse k-max optimization problem. We also prove that the inverse bottleneck problem is NP-hard. For the problem with the condition of independency of feasible subsets, we represent the cost function as a single variable function and prove its convexity. Then, we develop a polynomial time algorithm for the problem in Section 3. Some special cases with improved complexity are also discussed in this section.

2. PROBLEM SETTING AND COMPLEXITY

Let a ground set E of n elements be given and be equipped with a weight function $w : E \longrightarrow \mathbb{R}_+$ where $w(e) > 0$ for $e \in E$. A class of subsets $\mathcal{F} := \{F_0, F_1, \ldots, F_p\}$ with $F_i \subset E$, for $i = 0, 1, \ldots, p$, contains all feasible solution of a relevant combinatorial optimization problem. Furthermore, we assume that all subsets F_i have exactly m elements for $i = 0, 1, \ldots, p$. This assumption is applied in the case of cardinality constrained optimization, where all feasible sets have the similar cardinality. From now on, we index the elements in F_i such a way that $F_i := \{e^i_1, e^i_2, \ldots, e^i_m\}$ with $e^i_j \in E$ for $i = 0, 1, \ldots, p$ and $j = 1, \ldots, m$. Note that, two elements e^i_j and $e^{i'}_{j'}$ for $i \neq i'$ can stand for one element in E. Let $\sigma_i(\cdot)$ be a permutation such that

$$w(e^i_{\sigma_i(m)}) \leq w(e^i_{\sigma_i(m-1)}) \leq \ldots \leq w(e^i_{\sigma_i(1)}).$$

The k-max function $K : \mathcal{F} \longrightarrow \mathbb{R}_+$ is defined as $K(F_i) := w(e^i_{\sigma_i(k)})$ for $F_i \in \mathcal{F}$, i.e., it is the the weight of the k^{th} largest element in the underlying subset. In the function K, the value k is implicitly given for the sake of simplicity. A subset F_{i_0} is, by definition, a minimum k-max in $\mathcal{F}$ if and only if

$$K(F_{i_0}) \leq K(F_i)$$

for $i = 0, 1, \ldots, p$. Particularly, if $k = 1$, one obtains the so-called classical bottleneck element in $\mathcal{F}$.

We derive the condition for a set that is a minimum k-max of $\mathcal{F}$.

Theorem 1. *(k-max criterion) The subset F_{i_0} for $i_0 \in \{1, \ldots, p\}$ is a k-max in $\mathcal{F}$ iff*

$$\left|\Big\{e \in E : w(e) < K(F_{i_0})\Big\} \bigcap F_i\right| \leq m - k$$

for $i = 0, 1, \ldots, p$.

Proof. For a minimum k-max F_{i_0} in $\mathcal{F}$, then $K(F_{i_0}) \leq K(F_i)$ for all $i = 0, 1, \ldots, p$. Hence, there are at most $m-k$ elements in F_i, $i \in \{0, 1, \ldots, p\}$, whose weights are strictly less than $K(F_{i_0})$ as otherwise, the k-max objective at F_i is strictly smaller than $K(F_{i_0})$ and it contradicts the optimality of F_{i_0}. In other words, we obtain

$$\left|\Big\{e \in E : w(e) < K(F_{i_0})\Big\} \bigcap F_i\right| \leq m - k$$

for all $i = 0, 1, \ldots, p$.

Conversely, we assume that

$$\left|\Big\{e \in E : w(e) < K(F_{i_0})\Big\} \bigcap F_i\right| \leq m - k$$

for all $i = 0, 1, \ldots, p$. Then, there are at most $m-k$ elements in each $F_i \in \mathcal{F}$ whose weights are strictly less than $K(F_{i_0})$. The remaining k elements in F_i are larger than or equal to $K(F_{i_0})$, or $K(F_i) \geq K(F_{i_0})$. Hence, the set F_{i_0} is a minimum k-max in $\mathcal{F}$. $\square$

Note that in Theorem 1, the optimality criterion is equivalent to

$$\left|\Big\{e \in E : w(e) \geq K(F_{i_0})\Big\} \bigcap F_i\right| \geq k$$

for all $i = 0, 1, \ldots, p$. By taking $k = 1$, the k-max problem is indeed the bottleneck problem. The corresponding optimality criterion can be restated as

$$\left|\Big\{e \in E : w(e) \geq K(F_{i_0})\Big\} \bigcap F_i\right| \geq 1$$

for all $i = 0, 1, \ldots, p$, i.e., the set of elements in E whose weights are at least $K(F_{i_0})$ intersects all feasible solutions F_i for $i = 0, 1, \ldots, p$. The set $\Big\{e \in E : w(e) \geq K(F_{i_0})\Big\}$ is called a *cut* in E in term of Guan and Zhang [24]. Also, they proved that the subset F_{i_0} is a bottleneck if and only if the number of element(s) in the cut with respect to the maximum weight of elements in F_{i_0} is at least 1. Therefore, Theorem 1 is a generalization of the bottleneck condition in the mentioned paper.

Now we formally state the inverse k-max optimization problem. Let F_0 be, without loss of generality, a predetermined subset in $\mathcal{F}$. We can either increase or reduce each element e in E by $p(e)$ or $q(e)$, i.e., the perturbed weights are $\widetilde{w}(e) := w(e) + p(e) - q(e)$ for $e \in E$. The new weights are assumed to be positive, i.e., $q(e) < w(e)$ for $e \in E$. Furthermore, increasing or reducing one unit weight of e pays a cost $c(e)$. We denote $\widetilde{K}(\cdot)$ the k-max function with respect to the new weights. The inverse k-max optimization problem is to modify the elements in E in order to adapt the following conditions.

i) The set F_0 becomes a minimum k-max in $\mathcal{F}$ with respect to $\widetilde{w}$, i.e., $\widetilde{K}(F_0) \leq \widetilde{K}(F_i)$ for all $i = 1, \ldots, p$.

ii) Modifying cost $\sum_{e \in E} c(e)(p(e) + q(e))$ should be minimized.

ii) Bound constraints hold, i.e., $0 \leq p(e) \leq \bar{p}(e)$ and $0 \leq q(e) \leq \bar{q}(e)$ for $e \in E$.

We start with the complexity of the problem in the following result.

Theorem 2. *The inverse bottleneck optimization problem is NP-hard.*

Proof. Let us first revisit the set cover (SC) problem: 'Given a ground set $S = \{1, \ldots, n\}$ of n elements, its m subsets $S_1, S_2, \ldots, S_m$ with $\bigcup_{i=1}^m S_i = S$, and an integer l, determine whether there exists l of the given subsets such that their union equals S.' The (SC) is NP-complete according to Garey and Johnson [28].

The decision version of inverse bottleneck optimization problem (IBP) can be stated as: "Given an instance of the inverse bottleneck problem, determine if there exists a feasible modification such that the total cost is at most B."

Let an instance of (SC) be given. Let $e_1, e_2, \ldots, e_m$ represent m subsets $S_1, S_2, \ldots, S_m$. We derive an instance of (IBP) as in polynomial time as below:

- There are $n + 1$ feasible sets $F_0, F_1, F_2, \ldots, F_n$. If i in S_j, then element e_i is in F_j. To assure that $|F_0| = |F_j| = m$ for all $j = 1, \ldots, n$, we add auxiliary elements to each feasible set. Here, an element $e \in F_j \backslash \{e_1, e_2, \ldots, e_m\}$, then $e \notin F_s$ for $s \neq j$.
- We can choose elements in F_0 such that $K(F_0) = 2$. For F_j with $j \neq 0$, we set $w(e) = 1$ for all e in F_j and $j = 1, \ldots, m$.

- Set $\bar{p}(e_i) = 1$, $\bar{q}(e_i)$ for $i = 1, \ldots, n$ and $\bar{p}(e) = \bar{q}(e) = 0$ for any $e \notin \{e_1, e_2, \ldots, e_m\}$.
- We consider the uniform cost for modifying any element of the corresponding ground set $E = \bigcup_{i=0,\ldots,n} F_i$ and $B = l$.

After constructing an instance of the (IBP), we now prove that the (SC) and the (IBP) are equivalent.

If (SC) gives a 'yes' answer, there exists l subsets, without loss of generality, say $S_1, \ldots, S_l$ such that $\bigcup_{i=1}^{l} S_i = S$. We set $p(e_j) = \bar{p}(e_j) = 1$ for $j = 1, \ldots, l$. As $\bigcup_{i=1}^{l} S_i = 1, \ldots, n$, we know that there exists an element in $\{e_1, e_2, \ldots, e_m\}$ that is also in F_j for $j = 1, \ldots, n$. Furthermore, $\widetilde{K}(F_j) = 2$ for $j = 1, \ldots, n$ due to $\widetilde{w}(e_i) = 2$ for $i = 1, \ldots, m$. Therefore, F_0 is become the bottleneck of the modified instance and the cost is l.

If the answer to (IBP) is 'yes', then $\widetilde{K}(F_j) = 2$ for $j = 1, \ldots, n$ and the cost is at most l. For any element e_i for $i = 1, \ldots, m$ with $0 < p(e_i) < 1$, we can assume that $p(e_i) = 0$ without increasing the cost. Hence, we can assume that $p(e_i) \in \{0, 1\}$ for $i = 1, \ldots, m$. Let $\mathcal{J} := \{i \in 1, \ldots, m : p(e_i) = 1\}$. We know that $|\mathcal{J}| \leq l$ as $\sum_{i=1}^{m} p(e_i) \leq l$. Furthermore, each F_j contains an element e_i such that $p(e_i) = 1$ to ensure that $\widetilde{K}(F_j) = 2$ for $j = 1, \ldots, n$. Therefore, we imply that $\bigcup_{i \in \mathcal{J}} S_i = S$. We also attain 'yes' answer to (SC). $\square$

Corollary 3. *The inverse k-max optimization problem is NP-hard.*

3. SOME POLYNOMIALLY SOLVABLE CASES

By the proof Theorem 2, the overlap in the sets is a proven reason for NP-hardness. In this section, we focus on the special case with independent feasible subsets $F_0, F_1, \ldots, F_n$, i.e., the condition $F_i \cap F_j = \emptyset$ holds for any $i \neq j$ and $i, j \in \{1, \ldots, n\}$. Moreover, it is possible that $F_j \cap F_0 \neq \emptyset$ for some index $j \in \{1, \ldots, p\}$. Then, we know that $mp = O(n)$.

3.1. Special Properties of the Problem

We assume $K(F_0) > K(F_i)$, or equivalently

$$\left| \left\{ e \in E : w(e) < K(F_0) \right\} \bigcap F_i \right| > m - k,$$

for some indices $i \in \{1, \ldots, p\}$. Otherwise, a trivial solution with no modification is derived. Let the set

$$\mathcal{I} := \{i \in 1, \ldots, p : K(F_i) < K(F_0)\}$$

contain all indices that violate the optimality criterion. We parameterize the inverse k-max problem by setting $t := \widetilde{K}(F_0)$. This means t is the modified k-max objective of F_0. Moreover, we denote by

$$\mathcal{I}(t) := \{i \in \mathcal{I} : K(F_i) < t\}$$

the set of all feasible solutions whose k-max objectives are strictly less than t. To get the criterion in Theorem 1, we reduce the cardinality of the set $\Big\{e \in E : w(e) < t\Big\} \bigcap F_i$ for $i \in \mathcal{I}(t)$ until it is at most $m-k$. If $t > K(F_0)$, then $\Big|\Big\{e \in E : w(e) < t\Big\} \bigcap F_i\Big|$ is even larger than that of the current value $t_0 := K(F_0)$. Thus, we focus on the value $t < K(F_0)$ and increase the weights of elements e in F_i with $w(e) < t$ for $i \in \mathcal{I}$ to obtain the optimality criterion. The following result helps to observe the property of modifications.

Proposition 4. *If $t \in [w(e^0_{\sigma_0(j+1)}), w(e^0_{\sigma_0(j)}))$ for $j \geq k$, then there exist $j-k+1$ elements in the set*

$$\{e^0_{\sigma_0(j)}, e^0_{\sigma_0(j-1)}, \ldots, e^0_{\sigma_0(1)}\}$$

whose weights are indeed reduced. If $t \in [w(e^i_{\sigma_i(l+1)}), w(e^i_{\sigma_i(l)}))$ for $i \in \mathcal{I}(t)$, then there exist $k-l$ elements in the set

$$\{e^i_{\sigma_i(m)}, e^i_{\sigma_i(m-1)}, \ldots, e^i_{\sigma_i(l+1)}\}$$

whose weights are indeed increased.

Proof. As $t \in [w(e^0_{\sigma_0(j+1)}), w(e^0_{\sigma_0(j)}))$ and $t = \widetilde{K}(F_0)$, we do not modify the weights of elements in

$$\{e^0_{\sigma_0(m)}, e^0_{\sigma_0(m-1)}, \ldots, e^0_{\sigma_0(j-1)}\}.$$

On the other hand, as $w(e^0_{\sigma_0(r)}) > t$ for all $r = 1, 2, \ldots, j$ we have to reduce the weight of $j-k+1$ elements in

$$\{e^0_{\sigma_0(j)}, e^0_{\sigma_0(j-1)}, \ldots, e^0_{\sigma_0(1)}\}$$

to obtain the k-max value t.

For $i \notin \mathcal{I}(t)$, we know $K(F_i) \geq t$ and we hence do not increase the k-max objective value of F_i. Let us consider $i \in \mathcal{I}(t)$ and $t \in [w(e^i_{\sigma_i(l+1)}), w(e^i_{\sigma_i(l)}))$. We modify the weights of elements in F_i so that

$$\Big|\Big\{e \in E : \widetilde{w}(e) < t\Big\} \bigcap F_i\Big| = k-1.$$

Therefore, we increase exactly $k-l$ elements in

$$\{e^i_{\sigma_i(m)}, e^i_{\sigma_i(m-1)}, \ldots, e^i_{\sigma_i(l+1)}\}$$

in order to obtain the $(k-l)+l = k$ maximum objective t. $\square$

For example, we are given a ground set $E = \{e_1, \ldots, e_9\}$ with $w(e_i) = i$ for $i = 1, \ldots, 9$. Moreover, the feasible sets are $F_0 := \{e_1, e_2, e_7, e_8, e_9\}$ and $F_1 := \{e_2, e_3, e_4, e_5, e_6\}$ with $m = 5$. We consider $k = 2$, then $K(F_0) = 8$ and $\Big|\Big\{e \in E : w(e) < K(F_0)\Big\} \bigcap F_1\Big| = 5 > m-k$. Thus, F_0 is not a minimum k-max in $\mathcal{F}$. For $t \in [5, 6) = [w(e_2), w(e_7)) \cap [w(e_5), w(e_6))$, then $j = 3$ and $l = 1$. We reduce the weights of $j-k+1=2$ elements in $\{e_7, e_8, e_9\}$ and increase the weight of $k-l=1$ element in $\{e_2, e_3, e_4, e_5\}$ in order to obtain the k-max value t of both F_0 and F_1.

Next we identify an interval that contains feasible parameters of the inverse k-max optimization problem. After setting $\underline{w}(e_j^0) := w(e_j^0) - \bar{q}(e_j^0)$ for $j = 1, \ldots, m$, we obtain the set $\mathcal{M} := \{\underline{w}(e_j^0) : e_j^0 \in F_0\}$. Let $t_{\mathcal{M}}$ be the k^{th} maximum element in $\mathcal{M}$. This point can be found in $O(m)$ time by an advanced computation that prunes a half of the elements in each iteration, for example, one can see Balas and Zemel [29]. We know that the value of $\widetilde{K}(F_{i_0})$ can be reduced to at least $t_{\mathcal{M}}$. Moreover, as it is enough to reduce $\widetilde{K}(F_{i_0})$ to $\min_{i\in\mathcal{I}} K(F_i)$ to reach the optimality criterion, we set the lower bound of the parameter as

$$\underline{t} := \max\{t_{\mathcal{M}}; \min_{i\in\mathcal{I}} K(F_i)\}.$$

Similarly, after setting $\bar{w}(e_j^i) := w_j^i + \bar{p}(e_j^i)$ for $i = 1, \ldots, p$ and $j = 1, \ldots, m$, we consider the set $\mathcal{N}^i := \{\bar{w}(e_j^i) : e_j^i \in F_i\}$. Let $t_{\mathcal{N}^i}$ be the k^{th} largest element in $\mathcal{N}^i$. As it is enough to increase the k-max value of all F_i for $i = 1, \ldots, p$ to $K(F_0)$ such that conditions in Theorem 1 holds, we get the upper bound of the parameter

$$\bar{t} := \min\{K(F_0); \min_{i=1,\ldots,p} t_{\mathcal{N}^i}\}.$$

Computing $\underline{t}$ and $\bar{t}$ costs $O(pm) = O(n)$ time as there are at p feasible solutions in $\mathcal{F}\backslash\{F_0\}$.

Proposition 5. *The inverse k-max optimization problem is feasible iff* $\underline{t} \leq \bar{t}$.

Proof. If $\underline{t} > \bar{t}$, by Proposition 4 we can not modify the weights such that the optimality criterion holds. Conversely, if $\underline{t} \leq \bar{t}$, we guarantee to modify the weights to obtain the optimality criterion. $\square$

We now aim to represent the cost function. Considering

$$t \in [w(e^0_{\sigma_0(j+1)}); w(e^0_{\sigma_0(j)})) \bigcap_{i\in\mathcal{I}(t)} [w(e^i_{\sigma_i(l_i+1)}); w(e^i_{\sigma_i(l_i)})),$$

the cost function, due to Proposition 4, is written as

$$C(t) := \sum_{i\in\alpha(t)} c(e_i^0)(w(e_i^0) - t) + \sum_{i\in\mathcal{I}(t)} \sum_{j\in\beta^i(t)} c(e_j^i)(t - w(e_j^i)).$$

Here, $\alpha(t)$ is the set of the smallest $j-k+1$ elements in $\{c(e_i^0)(w(e_i^0)-t) : w(e_i^0) > t\}$ and $\beta^i(t)$ is the set of $k-l_i$ smallest elements in $\{c(e_j^i)(t - w(e_j^i)) : w(e_j^i) < t\}$. Note that $\alpha(t) \cap \beta^i(t) = \emptyset$ for $i \in \mathcal{I}(t)$. The cost $C(t)$ is indeed piecewise linear and one can present the corresponding slope at t as

$$slope(t) := \sum_{i\in\mathcal{I}(t)} \sum_{j\in\beta^i(t)} c(e_j^i) - \sum_{i\in\alpha(t)} c(e_i^0).$$

We can further derive an important property of the cost function.

Proposition 6. *The function $C(t)$ is convex for $t \in [\underline{t}, \bar{t}]$.*

Proof. Take $t, t' \in [\underline{t}; \bar{t}]$ and $t' > t$. We further assume that $t \in [w(e^0_{\sigma_0(j+1)}); w^0_{\sigma_0(j)})$ and $t' \in [w(e^0_{\sigma_0(j'+1)}); w^0_{\sigma_0(j')})$. As $t' > t$, one implies $j' \leq j$. We know that $\alpha(t)$ ($\alpha(t')$) is the set of the smallest $j-k+1$ ($j'-k+1$) elements in $\{c(e^0_i)(w(e^0_i)-t) : w(e^0_i) > t\}$ ($\{c(e^0_i)(w(e^0_i) - t') : w(e^0_i) > t'\}$). As $j' - k + 1 \leq j - k + 1$ and $\{e^0_i \in F_0 : w(e^0_i) > t'\} \subset \{e^0_i \in F_0 : w(e^0_i) > t\}$, there are two cases $\alpha(t') \subset \alpha(t)$ or $\alpha(t') \not\subset \alpha(t)$. In the first case, it holds $\sum_{i\in\alpha(t')} c(e^0_i) \leq \sum_{i\in\alpha(t)} c(e^0_i)$. For the second case, we take $c(e^0_j)(w(e^0_j) - t')$ in $\alpha(t')\backslash\alpha(t)$. As $c(e^0_{j'})(w(e^0_{j'}) - t')$ is a decreasing function, we obtain $c(e^0_{j'})(w(e^0_{j'}) - t') \leq c(e^0_{j'})(w(e^0_{j'}) - t)$. As $c(e^0_{j'})(w(e^0_{j'}) - t) \notin \alpha(t)$ and $j' - k + 1 \geq j - k + 1$, there exists an element $c(e^0_j)(w(e^0_j) - t) \in \alpha(t)$ that replaces $c(e^0_{j'})(w(e^0_{j'}) - t')$. Then, we imply that $c(e^0_{j'}) < c(e^0_j)$. For example, we consider Figure 1 with $c(e^0_j)(w(e^0_j) - t) = 2(4 - t)$ and $c(e^0_{j'})(w(e^0_{j'}) - t') = 3(3 - t)$. These two functions intersect at $t = 1$. Hence, for $t < 1$ and $t' > 1$, the ordering of these two functions is changed.

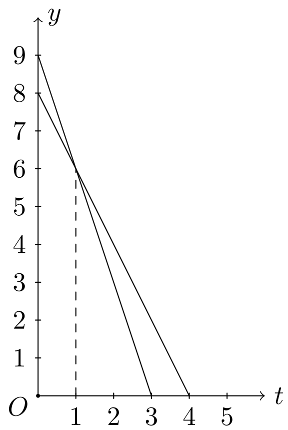

Figure 1: Two functions 3(3-t) and 2(4-t)

Therefore, we also get $\sum_{i\in\alpha(t')} c(e^0_i) \leq \sum_{i\in\alpha(t)} c(e^0_i)$ in the second case.

By the similar argument, we further get $\sum_{j\in\beta^i(t)} c(e^i_j) \geq \sum_{j\in\beta^i(t')} c(e^i_j)$ for $i \in \mathcal{I}(t)$. As $\mathcal{I}(t) \supset \mathcal{I}(t')$, the inequality *slope*$(t) \leq$ *slope*(t') holds for $t < t'$. In summary, the function $C(t)$ is indeed a convex function. $\square$

Next we aim to solve the inverse k-max optimization problem with efficient algorithm as well as discuss the complexity improvement in its special cases.

3.2. Solution Approach

We develop in this section a combinatorial algorithm for the inverse k-max optimization problem based on characteristics of the problem, which are settled in the previous section. As the function $C(t)$ is piecewise linear and convex for

$t \in [\underline{t}; \overline{t}]$, we can find an optimal solution on the set of its breakpoints in the following two steps.

Step 1: Find an interval that contains an optimal value.

We denote by

$$\mathcal{B} := \Big(\{w(e^i_j) : j = 1, \ldots, m \text{ and } i = 0, 1, \ldots, p\} \cup \{\underline{t}, \overline{t}\}\Big) \cap [\underline{t}; \overline{t}].$$

As there are at most $O(n)$ elements in $\mathcal{B}$, we can sort all items in $\mathcal{B}$ nondecreasingly in $O(n \log n)$ time and get $\mathcal{B} = \{b_1, b_2, \ldots, b_q\}$ with $b_1 \leq b_2 \leq \ldots \leq b_q$ and $q = O(n)$. Then, we search for an interval $[b_{i_0}, b_{i_0+1}]$ for $i_0 \in \{1, \ldots, q-1\}$, which contains an optimal solution of cost function $C(t)$ for $t \in [\underline{t}; \overline{t}]$ by a binary search algorithm. We start with computing $C(b_i)$ and $C(b_{i+1})$ for $i := \lfloor \frac{1+q}{2} \rfloor$. If $C(b_i) < C(b_{i+1})$, we know that the optimal solution is less than or equal b_i. Thus, we consider the new set $\mathcal{B} := \{b_1, \ldots, b_i\}$. Otherwise, we take into account the set $\mathcal{B} := \{b_{i+1}, \ldots, b_q\}$. We stop if the set $\mathcal{B}$ contains only one element; otherwise, we continue with the updated set $\mathcal{B}$.

For the complexity of this step, we consider the time for computing the cost $C(t)$ of modifying the set F_0 and F_i such that $\widetilde{K}(F_0) = \widetilde{K}(F_i) = t$ for $i \in \mathcal{I}(t)$ and $t \in \mathcal{B}$. Assume that $t = w(e^0_{\sigma_0(j+1)})$, the cost concerning modifying elements in F_0 can be done by finding and summing up $j - k + 1$ smallest elements of the set

$$\{c(e^0_{\sigma_0(l)})(w(e^0_{\sigma_0(l)}) - t) : l = 1, 2, \ldots, j\}$$

in $O(m)$ time. By the same argument, the cost for modifying elements in F_i can be also computed in $O(m)$ time. Thus, we can claim that the mentioned task can be computed in $O(pm) = O(n)$ time. As the binary search stop in $O(\log n)$ time, the complexity of this step is $O(n \log n)$ time. Note that, the advantage from this step is that the set $\mathcal{I}(t)$ does not change for $t \in [b_{i_0}, b_{i_0+1}]$. Hence, we know exactly the sets F_i whose k-max values are increased for $i \in \mathcal{I}(b_{i_0+1})$.

Step 2: Search for an optimal solution in $[b_{i_0}, b_{i_0+1}]$.

After completing **Step 1**, we aim to find the optimal solution of the problem in the predetermined interval. We first find the intersections of $c(e^i_j)(w(e^i_j) - t)$ and $c(e^i_{j'})(w(e^i_{j'}) - t)$ for all pairs $e^i_j, e^i_{j'}$ in the feasible set F_i with $j, j' \in \{1, \ldots, m\}$, $j \neq j'$, and $w(e^i_j), w(e^i_{j'}) \in [b_{i_0}, b_{i_0+1}], w(e^i_j) \neq w(e^i_{j'})$. They are calculated as

$$t^i_{jj'} := \frac{c(e^i_j)w(e^i_j) - c(e^i_{j'})w(e^i_{j'})}{c(e^i_{j'}) - c(e^i_j)}.$$

This procedure can be done in $O(n^2)$ time as there are at most $O(n^2)$ pairs of corresponding functions by considering the pair of functions based on elements in E. Then, it yields the set

$$\mathcal{Q} := \{t^i_{jj'} : i = 0, \ldots, p \text{ and } j \neq j', j, j' \in \{1, \ldots, m\}\} \cap [b_{i_0}, b_{i_0+1}].$$

Note that $\mathcal{Q}$ contains $O(n^2)$ many elements. In the previous section, we state that the presentation of $C(t)$ depends on the ordering of the relevant functions $c(e)(w(e) - t)$ for $e \in E$. Therefore, $\mathcal{Q}$ contains possible breakpoints of $C(t)$. As the objective function $C(t)$ is a piecewise linear convex function with breakpoints in $\mathcal{Q}$, we can find its optimal solution in $O(n^2 \log n)$ time by a binary search algorithm.

Due to the analysis in the two steps, we get the relative result.

Theorem 7. *The inverse k-max optimization problem with disjoint feasible subsets is solvable in $O(n^2 \log n)$ time with n being the input size of E.*

Let us illustrate the algorithm in the following example.

Example 8. *Consider the ground set $E := \{e_1, e_2, \ldots, e_{10}\}$. The set $\mathcal{F}$ of feasible solutions contains $F_0 := \{e_5, e_6, e_7, e_8, e_{10}\}$, $F_1 := \{e_1, e_3, e_4, e_8, e_9\}$, $F_2 := \{e_2, e_5, e_6, e_7, e_{10}\}$. The weights, costs, modification bounds of elements are given in Table 1.*

Table 1: Input data concerning of the inverse k-max problem

i	1	2	3	4	5	6	7	8	9	10
$w(e_i)$	1	1	2	2	8	8	9	9	10	10
$c(e_i)$	1	1	2	2	4	4	6	6	1	2
$\bar{p}(e_i)$	7	7	5	5	2	2	1	1	1	1
$\bar{q}(e_i)$	1	1	1	1	4	4	4	4	6	6

For $k = 3$, we get $K(F_0) = 9; K(F_1) = 2; K(F_2) = 8$. Hence, F_0 is not a 3-max of $\mathcal{F}$. We can compute $\underline{t} := 4$, $\bar{t} := 9$, and the set of break points $\mathcal{B} := \{4; 8; 9\}$.
Step 1: *We find the interval $[4, 8]$ which contains an optimal solution.*
Step 2: *We now consider the variable $t \in [4; 8]$. The set of break points in the interval $[4, 8]$ is $\mathcal{Q} := \{4; \frac{22}{7}; 8\}$, where $\frac{22}{7}$ is the intersection of two functions $c(e_5)(w(e_5) - t)$ and $c(e_{10})(w(e_{10}) - t)$. Furthermore, applying a binary search algorithm on $[4, 8]$ yields an optimal solution $t = 8$ with cost value $C(8) = 11$ (by setting $q(e_{10}) = 2$ and $p(e_1) = 7$).*

3.3. Improvement for Special Situations

Uniform-cost inverse *k*-max problem

We now consider the uniform-cost inverse k-max optimization problem, i.e., $c(e) = 1$ for all $e \in E$. For $t \in [w(e^0_{\sigma_0(j+1)}); w(e^0_{\sigma_0(j)}))$, we remind that there exist exactly $j - k + 1$ elements in

$$\left\{e^0_{\sigma_0(j)}, e^0_{\sigma_0(j-1)}, \ldots, e^0_{\sigma_0(1)}\right\}$$

whose weights are indeed reduced; see Proposition 4. We can directly find the desired elements for modification by gradually comparing

$$\underline{w}(e^0_{\sigma_0(j)}), \underline{w}(e^0_{\sigma_0(j-1)}), \ldots, \underline{w}(e^0_{\sigma_0(1)})$$

with t in that ordering and take the one with value less than t until there are $j-k+1$ elements. This procedure trivially derives a minimum cost for modifying elements in F_0 such that $\widetilde{K}(F_0) = t$ in $O(m)$ time. For example, we consider the set $E := \{e_i : i = 1, \ldots, 6\}$ with $w(e_i) = i$ for $i = 1, \ldots, 6$ and $F_0 = \{e_1, e_2, e_3, e_4\}$ with $\underline{w}(e_2) = \underline{w}(e_4) = 1$ and $\underline{w}(e_3) = 2$. For $k = 2$ and $t \in [1; 2)$, we reduce the weights of $j - k + 1 = 2$ elements in $\{e_2, e_3, e_4\}$. Then, the weights of e_2 and e_4 are indeed reduced.

For $i \in \mathcal{I}(t)$, we can find the minimum cost for $\widetilde{K}(F_i) = t$ in $O(m)$ time by the similar argument. Hence, it costs $O(pm) = O(n)$ time to compute the cost function at a fixed parameter t. After computing the set $\mathcal{B}$ as in the previous section, where $|\mathcal{B}| = O(n)$, we know that the $\mathcal{B}$ is the set of breakpoints of $C(t)$ since two functions $w(e^i_j)-t$ and $w(e^i_{j'})-t$ for $i \in \{0, \ldots, p\}$ and $j \neq j', j, j' \in \{1, \ldots, m\}$ are coincided or have no intersection. Therefore, we can apply a binary search approach to find an optimal solution of $C(t)$ in $\mathcal{B}$. Since the binary search terminates after $O(\log n)$ iterations, the time complexity of the problem is $O(n \log n)$.

Theorem 9. *The uniform-cost inverse k-max optimization with disjoint feasible subsets is solvable in $O(n \log n)$ time.*

Note that, for the uniform-cost problem, **Step 2** of the general problem is pruned. This simplification leads to the improvement in the computational complexity.

Inverse bottleneck optimization problem

Let us further consider another special case with $k = 1$, i.e., the corresponding problem is the inverse bottleneck optimization problem with the set of feasible solutions being a part of input. For $t \in [w(e^0_{\sigma_0(j+1)}); w(e^0_{\sigma_0(j)}))$, all elements in

$$\left\{e^0_{\sigma_0(j)}, e^0_{\sigma_0(j-1)}, \ldots, e^0_{\sigma_0(1)}\right\}$$

are indeed reduced as $j - k + 1 = j$ and there is exactly one element in F_i, for $i \in \mathcal{I}(t)$. By elementary computation, the slope of the cost function $C(t)$ is written as

$$\mathit{slope}(t) := \sum_{i \in \mathcal{I}(t)} \sum_{j \in \gamma^i(t)} c(e^i_j) - \sum_{l=1}^{j} c(e^0_{\sigma_0(l)})$$

Here, we use the abbreviation

$$\gamma^i(t) := \arg\min \left\{c(e^i_j)(t - w(e^i_j)) : j = 1, \ldots, m\right\}.$$

As the set $\gamma^i(t)$ contains exactly one index, we focus on identifying the lower envelope of a class of functions $\left\{c(e^i_j)(t - w(e^i_j)) : j = 1, \ldots, m\right\}$ and retrieve the desired index in each iteration. Applying the technique of Hershberger [30], we can find the lower envelope of m lines in $O(m \log m)$ time. Hence, it costs $O(n \log m)$ time to find all breakpoints corresponding to sets F_i for $i \in \mathcal{I}$. As there are n elements in E that correspond to n line segments, at most $O(n)$ breakpoints are concerned. The binary search algorithm stops in $O(\log n)$ time, we can thus solve the problem in $O(n \log n)$ time. To summarize, the final result is stated as below.

Theorem 10. *The inverse bottleneck problem with disjoint feasible subsets can be solved in $O(n \log n)$ time.*

Note that, we improve the complexity of this special case based on the computation of lower envelope of the cost modifications.

4. CONCLUSIONS

We considered the inverse k-max optimization problem, which is a generalization of the inverse bottleneck problem. It was showed that the problem is NP-hard in general. However, if the feasible solutions are disjoint, it can be solved in $O(n^2 \log n)$ time, where n is the cardinality of the ground set. Finally, we developed improved algorithms with $O(n \log n)$ time complexity for both inverse k-max optimization problem with uniform cost and the inverse bottleneck optimization problem. Studying this problem under various objective functions, for exammple, Chebyshev norms and Hamming distance, is a promising topic. Another new direction is to devise algorithmic approaches for the inverse k-max optimization problem with an intractable set of feasible solutions.

Acknowledgment. The authors would like to acknowledge the editor and the anonymous referees for their useful suggestions which helped to improve the paper.

Funding. This research received no external funding.

REFERENCES

[1] C. Heuberger, "Inverse combinatorial optimization: A survey on problems, methods, and results", *Journal of Combinatorial Optimization*, vol. 8, pp. 329-361, 2004.

[2] D. Burton, Ph.L. Toint, "On an instance of the inverse shortest paths problem", *Mathematical Programming*, vol.53, pp. 45-61, 1992.

[3] R.K. Ahuja, J.B. Orlin, *"Inverse optimization,"* Operations Ressearch, vol. 49, no. 5, pp. 771-783, 2001.

[4] M.C. Cai, C.W. Duin, X. Yang, J. Zhang, "The partial inverse minimum spanning tree problem when weight increase is forbidden", *European Journal of Operational Research*, vol. 188, no. 2, pp. 348-353, 2008.

[5] D.S. Hochbaum, "Efficient algorithms for the inverse spanning-tree problem", *Operations Research*, vol. 51, no. 5, pp. 785-797, 2003.

[6] B. Li, Z. Sheng, "On the inverse minimum spanning tree problem with minimum number of perturbed edges", *Journal of System Science and System Engineering*, vol. 12, pp. 350-359, 2003.

[7] R.K. Ahuja, J.B. Orlin, "Combinatorial algorithms for inverse network flow problems", *Networks*, vol. 40, no. 4, pp. 181-187, 2002.

[8] A. Deaconu, "The inverse maximum flow problem considering l_∞-norm", *RAIRO Operations Research*, vol. 402, no. 3, pp. 401-414 , 2008.

[9] M.C. Cai, Y. Li, "Inverse matroid intersection problem", *Mathematical Methods of Operations Research*, vol. 45, no. 4, 235-243, 1997.

[10] Z. Zhang, S. Li, H.J. Lai, D.Z. Du, "Algorithms for the partial inverse matroid problem in which weights can only be increased", *Journal of Global Optimization*, vol. 65, pp. 801-811, 2016.

[11] J. Roland, J.R. Figueira, Y.D. Smet, "The inverse $\{0, 1\}$-knapsack problem: Theory, algorithms and computational experiment", *Discrete Optimization*, vol.10, no. 2, pp. 181-192, 2013.

[12] Y. Chung, M. Demange, "On inverse traveling salesman problems", *4OR*, vol. 10, pp. 193-209, 2012.

[13] I. Keshtkar, M. Ghiyasvand, "Inverse quickest center location problem on a tree", *Discrete Applied Mathematics*, vol. 260, no. 15, pp. 188-202, 2019.

[14] V.H. Pham, K.T. Nguyen, T.T. Le, "Inverse stable point problem on trees under an extension of Chebyshev norm and Bottleneck Hamming distance", *Optimization Methods and Software*, vol. 36, pp. 755-772 2021.

[15] S. Taghikhani, F. Baroughi, B. Alizadeh, "A generalized interval type-2 fuzzy random variable based algorithm under mean chance value at risk criterion for inverse 1-median location problems on tree networks with uncertain costs", *Journal of Computational and Applied Mathematics*, vol. 408, 114104, 2022.

[16] M. Hasanzadeh, B. Alizadeh, F. Baroughi, "Optimal algorithms for some inverse uncapacitated facility location problems on networks", *Optimization Methods and Software*, vol. 37, no. 3, pp. 982-1005, 2022.

[17] X. Qian, X. Guan, J. Jia, Q. Zhang, P. M. Pardalos, "Vertex quickest 1-center location problem on trees and its inverse problem under weighted l_∞ norm", *Journal Global Optimization*, 2022. Online. doi: https://doi.org/10.1007/s10898-022-01212-5.

[18] T.H.N. Nhan, K.T. Nguyen, H. Nguyen-Thu, "The Minmax Regret Reverse 1-Median Problem on Trees with Uncertain Vertex Weights", *Asia-Pacific Journal of Operational Research*, 2022. Online. doi: https://doi.org/10.1142/S0217595922500336.

[19] S. Nickel, J. Puerto, *Location Theory - A unified approach*, Springer, 2005.

[20] E. Gassner, "An inverse approach to convex ordered median problems in trees", *Journal of Combinatorial Optimization*, vol. 23, no.2, pp. 261-273, 2012.

[21] K.T. Nguyen, A. Chassein, "The inverse convex ordered 1-median problem on trees under Chebyshev norm and Hamming distance", *European Journal of Operational Research*, vol. 247, no. 3, pp. 774-781, 2015.

[22] K.T. Nguyen, H. Nguyen-Thu, N.T. Hung, "On the complexity of inverse convex ordered 1-median problem on the plane and on tree networks", *Mathematical Methods of Operations Research*, vol. 88, no. 2, pp. 147-159, 2018.

[23] V.H. Pham, K.T. Nguyen, "Inverse anti-k-centrum problem on networks with variable edge lengths", *Taiwanese Journal of Mathematics*, vol. 24, no. 2, pp. 501-522, 2020.

[24] X. Guan, J. Zhang, "Inverse constrained bottleneck problems under weighted l_∞-norm", *Computers and Operations Research*, vol. 34, no. 11, pp. 3243-3254, 2007.

[25] X. Guan, Y. Cao, "Constrained and bicriteria inverse bottleneck optimization problems under weighted Hamming distance", *Optimization*, vol. 61, no. 2, pp. 1-14, 2012.

[26] X. Guan, J. Zhang, "Inverse constrained bottleneck problems on networks" In: Gervasi O. et al. (eds) *Computational Science and Its Applications – ICCSA 2005*. ICCSA 2005. Lecture Notes in Computer Science 3483. Springer, Berlin, Heidelberg, 2005.

[27] J. Gorski, S. Ruzika, "On k-max-optimization", *Operations Research Letters*, vol. 37, no. 1, pp. 23-26, 2009.

[28] M.R. Garey, D.S. Johnson, *Computers and intractability: A guide to the theory of NP-completeness*, W. H. Freeman and Co., Sanfrancisco, CA, 1979.

[29] E. Balas, E. Zemel, "An algorithm for large zero-one knapsack problems", *Operations Research*, vol. 28, no. 5, pp. 1130-1154, 1980.

[30] J. Hershberger, "Finding the upper envelope of n line segments in $O(n \log n)$ time", *Information Processing Letters*, vol. 33, no. 4, pp. 169-174, 1989.