Can Tho University Journal of Science *Vol. 54, No. 5 (2018): 72-76*

Can Tho University Journal of Science
website: sj.ctu.edu.vn

DOI: 10.22144/ctu.jen.2018.027

Inverse version of the k^{th} maximization combinatorial optimization problem

Huynh Duc Quoc[1*], Nguyen Trung Kien[2], Trinh Thi Cam Thuy[1], Ly Hong Hai[3] and Vo Ngoc Thanh[4]

[1]*Department of Mathematics, College of Natural Sciences, Can Tho University, Vietnam*

[2]*Department of Mathematics Education, School of Education, Can Tho University, Vietnam*

[3]*Foreign Language, Information Technology Center, Can Tho Medical College, Vietnam*

[4]*Hung Loi High School, Vietnam*

[*]*Correspondence: Huynh Duc Quoc (email: hdquoc@ctu.edu.vn)*

Article info.

Received 07 Dec 2017
Revised 20 Apr 2018
Accepted 20 Jul 2018

Keywords

Chebyshev norm, hamming distance, inverse optimization, maximization

ABSTRACT

A ground set of n elements and a class of its subsets, also known as feasible solutions, is given. Moreover, each element in the ground set is associated a positive weight. In the setting of the original combinatorial optimization problem, each feasible solution corresponds to an objective value, often measured under the sum or the max of all element weights in the underlying solution. This paper is to address the problem of modifying the weight of elements in the ground set such that a prespecified subset becomes the k^{th} maximizer with respect to new weights and the cost is minimized. This problem is called the inverse version of the k^{th} maximization combinatorial optimization. Two quadratic algorithms were developed to solve this problem with sum objective function under Chebyshev norm and the bottleneck Hamming distance. Additionally, if the objective function is the max function then this problem can be solved in $O(n^2 \log n)$ time.

Cited as: Quoc, H.D., Kien, N.T., Thuy, T.T.C., Hai, L.H. and Thanh, V.N., 2018. Inverse version of the k^{th} maximization combinatorial optimization problem. Can Tho University Journal of Science. 54(5): 72-76.

1 INTRODUCTION

In a combinatorial optimization problem, one often supposes that the parameters such as costs, capacities, profits, etc. are already known and aims to find an optimal solution. However, in many real-life situations, the estimation or approximation for the parameters are known and it is difficult to find the exact optimizer. The fundamental idea of inverse (combinatorial) optimization problem (Heuberger, 2004) is to change the parameters of the corresponding original problem such that a predetermined solution becomes optimal with respect to new parameters and the cost function is minimized.

The inverse optimization problem was first investigated by Burton *et al*. (1992). They studied the inverse shortest path problem arising in seismic tomography and gave an application to forecast the movement of earthquakes. Since then, lots of useful applications in the reality of the inverse optimization problem have been proposed by many researchers. In 1995, the relation between the inverse shortest path and minimal cutset problem was proved by Xu *et al*. (1995). Then, Zhang *et al*. (1996) recommended strongly polynomial time algorithms to solve the inverse version of assignment and minimum cost flow problems. Two years later, Zhang *et al*. (1998) demonstrated that the inverse problem of minimum cuts can be transformed in a direct way into a minimum cost circulation problem, and therefore, can be solved successfully by strongly polynomial algorithms. In the same year, Hu *et al*. (1998) also designed an $O(n^3)$ algorithm to solve the inverse shortest path arborescence problem under l_1 norm. Recently, Nguyen *et al*. (2015) explored the inverse convex ordered 1-median problem on trees with the cost function considered under Chebyshev norm and the bottleneck Hamming distance. They established an $O(n^2 \log n)$ time algorithm to solve this problem.

In many practical situations, the k^{th} maximizer of a problem is focused. For example, as the price of transportations, services and travel agencies in the first classes are expensive or unreasonable, one had better to choose the ones in choosing the k^{th}-class. Therefore, it is necessary to justify parameters of a model so that the desired solution becomes the k^{th} best one. In this paper, the inverse k^{th} maximization problem under Chebyshev norm and bottleneck Hamming distance are studied. According to the best of our knowledge, the problem has not been under investigation so far. The objective function of the original problem is considered in two forms, viz., *sum* and *max*. For the sum function, the problem can be solved in quadratic time. Concerning the max function, an $O(n^2 \log n)$ algorithm is developed to solve the corresponding inverse problem.

This paper is organized as follows. Section 2 briefly recalls the combinatorial optimization and its inverse version. Two quadratic algorithms that solve the inverse k^{th} maximization problem are developed in Section 3. Finally, in Section 4, inverse k^{th} maximization optimization problem with max function is coined. It shows that the problem is solvable in $O(n^2 \log n)$ time.

2 PROBLEM DEFINITION

Given a ground set $G := \{e_1; e_2; \dots; e_n\}$ and let F be a class of subsets of G, i.e., $F := \{E_1; E_2; \dots; E_p\}$, where $E_i \subset G$, for $i = 1, \dots, p$. The set F is often considered as the set of all feasible solutions for a corresponding combinatorial optimization problem on G. Moreover, each element e_j is associated with a non-negative weight, say $w(e_j)$, for $j = 1, \dots, n$. The weight of E can be either measured by the sum of all elements in E, i.e.,

$$w_s(E) := \sum_{e \in E} w(e),$$

or by the max of its members, i.e.,

$$w_m(E) := \max_{e \in E} w(e).$$

A solution $E \in F$ is the k^{th} maximizer of the combinatorial optimization problem with respect to sum (max) function on F iff $w_s(E) = w_s(E_{(k)})$ $\left(w_m(E) = w_m(E_{(k)})\right)$ in the sorting

$$w_s(E_{(1)}) \le w_s(E_{(2)}) \le w_s(E_{(3)}) \le \cdots \le w_s(E_{(k)}) \le \cdots \le w_s(E_{(p)})$$

or

$$w_m(E_{(1)}) \le w_m(E_{(2)}) \le w_m(E_{(3)}) \le \cdots \le w_m(E_{(k)}) \le \cdots \le w_m(E_{(p)}).$$

Here, $(.)$ is a permutation on the set of $\{1,2, \dots, p\}$.

Example 2.1

Given a network in Fig. 1., the solution sets $F = \{E_1; E_2; E_3; E_4; E_5; E_6\}$ can be considered as the set of all paths connecting two leaves of the tree. Then the subsets in F is represented as follows:

$$E_1 = \{e_1; e_3; e_4\}, E_2 = \{e_2; e_3; e_4\}, E_3 = \{e_1; e_3; e_5\}, E_4 = \{e_2; e_3; e_5\}, E_5 = \{e_1; e_2\}, E_6 = \{e_4; e_5\}.$$

$w_s(E_1) = 11; w_s(E_2) = 12; w_s(E_3) = 13; w_s(E_4) = 14; w_s(E_5) = 5; w_s(E_6) = 10.$

$w_m(E_1) = 5; w_m(E_2) = 5; w_m(E_3) = 6; w_m(E_4) = 6; w_m(E_5) = 3; w_m(E_6) = 6.$

Choose $k = 4$, then E_2 is the 4^{th} maximizer with respect to sum function. Correspondingly, E_3 (or E_5, E_6) is the 4^{th} maximizer with respect to max function.

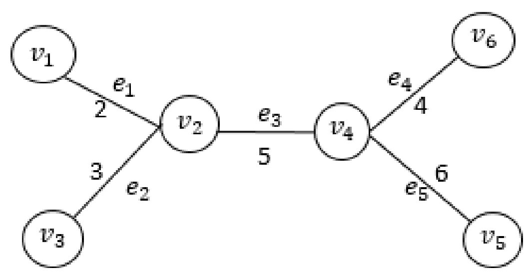

Fig. 1: An instance of a network

Given a set of feasible solutions F, a weight function $w(.)$ and a prespecified set $E^* \in F$. The weight of each element is modified by augmenting or reducing, i.e., $\tilde{w}(e) = w(e) + p(e) - q(e)$. The inverse k^{th} maximization is stated as follows:

E^* become the k^{th} maximization (corresponding sum or max function) with respect to new weights $\widetilde{w}$.

Cost function $f(p,q)$ is minimized.

Variables are in certain bounds, i.e., $0 \le p(e) \le \bar{p}(e)$ and $0 \le q(e) \le \bar{q}(e)$ for $e \in G$.

3 PROBLEM WITH SUM FUNCTION

3.1 Under Chebyshev norm

The cost function can be written as $f(p,q) = \max\{c^+(e)p(e), c^-(e)q(e)\}$, where $c^+(e)(c^-(e))$ is the cost to increase (decrease) one unit weight of an edge $e \in G$. Assume that $w_s(E^*) < w_s(E_{(k)})$, then E^* is not a k^{th}-max of the problem.

Proposition 3.1 *In the optimal solution of the inverse k^{th} max optimization problem, one increases the weights of elements in E^* and reduces the weights of elements in $G\backslash E^*$.*

By this proposition, we set $q(e) = 0, \forall e \in E^*$, and $p(e) = 0, \forall e \in G\backslash E^*$. Hence, we set

$$x(e) = \begin{cases} p(e), & \text{if } e \in E^* \\ q(e), & \text{if } e \in G\backslash E^* \end{cases} \quad \text{and} \quad \bar{x}(e) = \begin{cases} \bar{p}(e), & \text{if } e \in E^* \\ \bar{q}(e), & \text{if } e \in G\backslash E^*. \end{cases}$$

The weight of $e \in E^*$ ($e \in G\backslash E^*$) is said to be modified by an amount $x(e)$ if it is augmented (reduced) by $x(e)$. The cost is consequently written as

$$f(p,q) = \max_{e\in E}\{c(e)x(e)\}.$$

Here, $c(e) := c^+(e)$ if $e \in E^*$ and $c(e) := c^-(e)$ otherwise.

The following denotation is further introduced:

$$\Delta(E,E') = (E\backslash E') \cup (E'\backslash E); \delta(E,E') = w_s(E) - w_s(E').$$

Proposition 3.2 In the optimal solution of the problem, there exists at least one solution set $E \in F, w_s(E) > w_s(E^*)$, such that $\widetilde{w}_s(E) = \widetilde{w}_s(E^*)$.

<u>Proof.</u> For $E \in F, w_s(E) > w_s(E^*)$. Let us consider $\widetilde{w}_s(E) - \widetilde{w}_s(E^*) = \sum_{e\in E\cap E^*}(w(e) + x(e)) + \sum_{e\in E\backslash E^*}(w(e) - x(e)) - \sum_{e\in E^*}(w(e) + x(e))$

$$= w_s(E) - w_s(E^*) - \sum_{e\in\Delta(E,E^*)} x(e) = \delta(E,E^*) - \sum_{e\in\Delta(E,E^*)} x(e).$$

So, we only find $x(e), e \in \Delta(E,E^*)$ such that $\delta(E,E^*) - \sum_{e\in\Delta(E,E^*)} x(e) = 0$ for at least one E with $w_s(E) > w_s(E^*)$. Indeed, if $\widetilde{w}_s(E) - \widetilde{w}_s(E^*) > 0$, for all $E \in F$. Then, we cannot transform E^* into the k^{th} maximizer, there is nothing to discuss.

By this proposition, we consider the object at which there exists $E_{(i)}, w_s(E_{(i)}) > w_s(E^*)$ such that the modified weights of them are equal.

Let us consider the current objective value t, then $\widetilde{w}_s(E) - \widetilde{w}_s(E^*)$ is reduced as much as possible if

$$x(e) := x^t(e) = \begin{cases} \bar{x}(e), & \text{if } c(e)\bar{x}(e) \le t, \\ \dfrac{t}{c(e)}, & \text{otherwise.} \end{cases}$$

Hence, to search for the minimum value t s.t. $\widetilde{w}_s(E) - \widetilde{w}_s(E^*)$, we apply a binary search algorithm. Function $g(t) := \delta(E,E^*) - \sum_{e\in\Delta(E,E^*)} x^t(e)$ is a descending function with breakpoints in $\mathcal{B} := \{c(e)\bar{x}(e)\}_{e\in\Delta(E,E^*)} = \{t_1, t_2, \dots, t_j\}$.

We consider

$$\sum_{e\in\Delta(E,E^*)} x^t(e) = \sum_{e\in\rho(t)} \bar{x}(e) + \sum_{e\notin\rho(t)} \frac{t}{c(e)},$$

where, $\rho(t) := \{e: c(e)x(e) \le t\}$.

Then, we find the minimum value t_{i_0} such that $\sum_{e\in\Delta(E,E^*)} x^{t_{i_0}}(e) \ge \delta(E,E^*)$ in linear time. Therefore, the objective value t such that $\sum_{e\in\Delta(E,E^*)} x^t(e) = \delta(E,E^*)$ can be found by

$$\sum_{e:c(e)\bar{x}(e)\le t_{i_0-1}} \bar{x}(e) + \left(\sum_{e:c(e)\bar{x}(e)>t_{i_0-1}} \frac{1}{c(e)}\right)t = \delta(E,E^*).$$

Algorithm 1: Finds the minimum value $t_{i_0} \in \mathcal{B}$.

Input: An instance of the problem with $w_s(E^*) < w_s(E)$.

Find the set $\mathcal{B}$ and index its elements, sort it as increasing other $t_1 \le t_2 \le \cdots \le t_j$.

Set $a := 1, b := j$.

while $|\mathcal{B}| > 1$ **do**

Set $h \coloneqq \left\lfloor \frac{a+b}{2} \right\rfloor$, compute $g(t_h)$.

if $g(t_h) > 0$ **then**

Delete all elements in $\mathcal{B}$ which are smaller than t_h and set $a \coloneqq h + 1$.

else

Delete all elements in $\mathcal{B}$ which are larger than t_h and set $b \coloneqq h$.

end if

end while

Output: The remaining t_{i_0} in $\mathcal{B}$ such that $\sum_{e\in\Delta(E,E^*)} x^{t_{i_0}}(e) \geq \delta(E, E^*)$.

With two sets E_i, E_j, we can find optimal parameters t_i, t_j, respectively. It is clearly to see that if $t_i \leq t_j$ then the optimal objective value of the problem is at most $t \coloneqq \max\{t_i, t_j\}$ as $t < \alpha$ then $\widetilde{w}_s(E^*) < \widetilde{w}_s(E_i)$ or $\widetilde{w}_s(E^*) < \widetilde{w}_s(E_j)$.

Algorithm 2: Solves the inverse k^{th} max problem under Chebyshev norm

Input: An instance of the problem with $w_s(E^*) < w_s(E_{(k)})$.

Find all feasible solutions E, s.t. $w_s(E) > w_s(E^*)$.

Find all objective value such that $\widetilde{w}_s(E) = \widetilde{w}_s(E^*)$, denote by $C(E^*, E)$.

Sort all costs $\{C(E^*, E)\}$ as increasing order.

Output: Optimal cost $C(E^*, E)$.

Clearly, $C(E^*, E) = C(E^*, E_{(l)})$ with $l = k - r$, r is the number of E such that $w_s(E) < w_s(E^*)$. In the input data, the computation on F can be done in linear time. For example, $\Delta(E, E^*)$ can be computed in linear time by just scanning the elements in the two sets E and E^*. Furthermore, we can calculate $C(E, E_{(i)})$ in $O(|\Delta(E, E^*_{(i)})|)$ time for $i = 1, \ldots, j$. Therefore, it costs $\sum_{i=1}^{j} O(|\Delta(E, E^*_{(i)})|) = O(n^2)$ time to compute all required costs.

Theorem 3.1. *The inverse k^{th} max combinatorial optimization can be solved in $O(n^2)$ time, where n is the number of elements in the ground set.*

3.2 Under bottleneck Hamming distance

For this situation, the objective function can be written as follows:

$$f(p, q) = \max_{e\in E}\{c^+(e)H(p(e)), c^-(e)H(q(e))\},$$

where $H(.)$ is the Hamming distance as $H(\theta) = \begin{cases} 0, \text{if } \theta = 0 \\ 1, \text{if } \theta \neq 0. \end{cases}$

Similar to the case of Chebyshev norm, the weights of elements in E^* are augmented and the weights of others are reduced. Hence, we simplify the objective function as

$$f(x) = \max_{e\in E}\{c(e)H(x(e))\}.$$

We also get $\widetilde{w}_s(E) - \widetilde{w}_s(E^*) = \delta(E, E^*) - \sum_{e\in\Delta(E,E^*)} x(e)$ for $w_s(E) > w_s(E^*)$.

Same as above, $\widetilde{w}_s(E) - \widetilde{w}_s(E^*)$ is reduced as much as possible if

$$x(e) := \begin{cases} \bar{x}(e), \text{if } \ c(e) \leq t, \\ 0, \text{otherwise}. \end{cases}$$

Consider $\mathcal{B} := \{c(e)\}_{e\in\Delta(E,E^*)}$. The optimal cost obtains one value in $\mathcal{B}$. We can calculate the minimum value in $\mathcal{B}$ such that $\widetilde{w}_s(E) - \widetilde{w}_s(E^*) = 0$ in linear time by applying binary search algorithm. Hence, we also get the following result.

Theorem 3.2 *The inverse k^{th} max combinatorial optimization problem is solvable in quadratic time.*

4 PROBLEM WITH MAX FUNCTION

4.1 Under Chebyshev norm

We consider the weight function as $w_m(E) = \max_{e\in E}\{w(e)\}$ as the reader is used to the solution approach of this problem with sum function. It is based on the property of the difference $w_s(E) - w_s(E^*)$. Hence, for $w_m(E) > w_m(E^*)$, let us investigate

$$\begin{aligned} \widetilde{w}_m(E^*) - \widetilde{w}_m(E) &= \max_{e\in E^*}\{w(e) + x(e)\} \\ &\quad - \max\left\{\max_{e\in E^*\cap E}\{w(e) + x(e)\}, \max_{e\in E\setminus E^*}\{w(e) - x(e)\}\right\}. \end{aligned}$$

As we find the objective value such that $\widetilde{w}_m(E) = \widetilde{w}_m(E^*)$. For simplicity, we can denote by

$$D(E, E^*) = \max_{e\in E^*}\{w(e) + x(e)\} - \max_{E\setminus E^*}\{w(e) - x(e)\}.$$

We get the following proposition.

Proposition 4.1 *The inequality $D(E, E^*) \geq \widetilde{w}_m(E) - \widetilde{w}_m(E^*)$ is always hold for all x, and $D(E, E^*) = 0$ iff $\widetilde{w}_m(E) = \widetilde{w}_m(E^*)$.*

Proof. Obviously, $D(E,E^*) \geq \widetilde{w}_m(E) - \widetilde{w}_m(E^*)$ is always hold. If $D(E,E^*) = 0$ then $\max_{e\in E^*}\{w(e) + x(e)\} = \max_{E\setminus E^*}\{w(e) - x(e)\}$. Because $\max_{e\in E^*}\{w(e) + x(e)\} \geq \max_{E\cap E^*}\{w(e) - x(e)\}$, we get $\widetilde{w}_m(E^*) - \widetilde{w}_m(E) = 0$. Other way, if $\widetilde{w}_m(E) - \widetilde{w}_m(E^*) = 0$, for the least cost, we can find $\{x(e)\}_{e\in E\setminus E^*}$ such that $\max_{e\in E^*}\{w(e) + x(e)\} = \max_{E\setminus E^*}\{w(e) - x(e)\}$. Finally, we get the result $D(E,E^*) = 0$.

By this proposition, we can consider $D(E,E^*)$ instead of $\widetilde{w}_m(E) - \widetilde{w}_m(E^*)$. Let $\mathcal{B} := \{c(e)\bar{x}(e)\}_{e\in\Delta(E,E^*)}$. We sort all elements in $\mathcal{B}$ to get $\mathcal{B} = \{t_1, \ldots, t_j\}$ with $t_1 \leq t_2 \leq \cdots \leq t_j$. We apply a binary search algorithm to find minimum value $t_{i_0} (i_0 > 1)$ such that $D(E,E^*) > 0$ with respect to cost t_{i_0}. It can be done similarly to Algorithm 1. Hence, we further consider the function $D(E,E^*)$ for $t \in [t_{i_0-1}; t_{i_0}]$, i.e.,

$$D(E,E^*) = \max_{e:c(e)\bar{x}(e) > t_{i_0-1}}\left\{w(e) + \frac{t}{c(e)}\right\} - \max_{e:c(e)\bar{x}(e) > t_{i_0-1}}\left\{w(e) - \frac{t}{c(e)}\right\}.$$

Where at $t = t_{i_0-1}, D(E,E^*) < 0, D(E,E^*) = 0$ is obtained at

$$t^* = \operatorname{argmin} \max\left\{\max_{e\in E^*:c(e)\bar{x}(e) > t_{i_0-1}}\left\{w(e) + \frac{t}{c(e)}\right\}, \max_{e\in E\setminus E^*:c(e)\bar{x}(e) > t_{i_0-1}}\left\{w(e) - \frac{t}{c(e)}\right\}\right\}.$$

Hence t^* can be found in linear time by the algorithm of Gassner (2009).

Theorem 4.1 *The inverse k^{th} max combinatorial optimization problem with max function can be solved in $O(n^2 \log n)$ time.*

4.2 Problem under bottle-neck Hamming distance

We also consider the gap function

$$D(E,E^*) = \max_{e\in E^*}\{w(e) + x(e)\} - \max_{e\in E\setminus E^*}\{w(e) - x(e)\}.$$

Let $\mathcal{B} := \{t_1, \ldots, t_j\}$ with $t_1 \leq t_2 \leq \cdots \leq t_j$ as above. Then, we apply a binary search algorithm to find the minimum value such that $D(E,E^*) \geq 0$. The corresponding cost is $C(E)$.

Theorem 4.2 *The inverse k^{th} max combinatorial optimization problem with max function under bottle-neck Hamming distance can be solved in $O(n^2 \log n)$ time.*

5 CONCLUSION

We addressed the inverse k^{th} maximization problem with the sum and max function under Chebyshev norm and bottleneck Hamming distance. Based on a binary search algorithm, we developed algorithms that solved the underlying problem in quadratic time with sum function, and $O(n^2 \log n)$ with the other one. For future research, we will consider the inverse k^{th} maximization under various objective function, e.g., rectilinear norm or weighted sum Hamming distance.

REFERENCES

Burton, D., and Toint, P.L., 1992. On an instance of the inverse shortest paths problem. Mathematical Programming. 53(1): 45-61.

Gassner, E., 2009. Up-and downgrading the 1-center in a network. European Journal of Operational Research. 198(2): 370-377.

Heuberger, C., 2004. Inverse combinatorial optimization: A survey on problems, method, and results. Journal of Combinatorial Optimization. 8(3): 329-361.

Hu, Z., and Liu Z., 1998. A strongly polynomial algorithm for the inverse shortest arborescence problem. Discrete applied mathematics. 82(1-3): 135-154.

Nguyen, K.T., and Chassein, A., 2015. Inverse eccentric vertex problem on networks. Central European. Journal of Operations Research. 23(3):687-698.

Xu, S. and Zhang, J., 1995. An inverse problem of the weighted shortest path problem. Japan Journal of Industrial and Applied Mathematics. 12(1): 47-59.

Zhang, J., and Liu, Z., 1996. Calculating some inverse linear programming problems. Journal of Computational and Applied Mathematics. 72(2): 261-273.

Zhang, J. and Cai, M.C., 1998. Inverse problem of minimum cuts. Mathematical Methods of Operations Research. 47(1): 51-58.