# Inverse Multicast Quality of Service Routing Problem with Bandwidth and Delay Under the Weighted $l_1$ Norm

Longcheng Liu[1(✉)], Yu'an Chen[1], Wenhao Zheng[1], and Deqing Wang[2]

[1] School of Mathematical Sciences, Xiamen University, Xiamen 361005, People's Republic of China
longchengliu@xmu.edu.cn

[2] School of Information Science and Engineering, Xiamen University, Xiamen 361005, People's Republic of China

**Abstract.** Quality of Service (QoS) Routing problem has been attracting considerable attention thanks to the rapid development of the high-speed communication network, image processing and computer science. In the past decades, many Quality of Service Routing algorithms were presented based on the QoS requirements and the resource constraints.

The idea of the inverse optimization problem is to modify the given or estimated parameters such that the given feasible solution became an optimal solution. The modification costs are measured by different norms, such as $l_1$ norm, $l_2$ norm, $l_\infty$ norm, Hamming distance and so on.

In this paper, we consider the inverse multicast quality of service routing problems under the weighted $l_1$ norm. We present combinatorial algorithms which can be finished in strongly polynomial time.

**Keywords:** Communication network · Quality of service routing · Inverse problem · Strongly polynomial combinatorial algorithm

## 1 Introduction

The notion of Quality of Service has been proposed to capture the qualitatively or quantitatively defined performance contract between the service provider and the user applications in communication networks. The quality of service requirement of a connection is given as a set of constraints, which can be link constraints, path constraints, or tree constraints. The routing problems can be divided into two major classes: unicast routing and multicast routing.

The unicast quality of service routing problem with bandwidth and delay is defined as follows: given a graph $(V, E, s, t)$, $V = \{1, 2, \ldots, n\}$, $E = \{e_1, e_2,$

This research is supported by the National Natural Science Foundation of China (Grant No. 11001232, 61301098), Fujian Provincial Natural Science Foundation of China (Grant No. 2012J01021) and Fundamental Research Funds for the Central Universities (Grant No. 20720160035, 20720150002).


S. Guo et al. (Eds.): TridentCom 2016, LNICST 177, pp. 158–167, 2017.
DOI: 10.1007/978-3-319-49580-4_15

$\ldots, e_m\}$, $s$ is the service provider and $t$ is the terminal user. Let each edge $e_i$ has associated bandwidth $b_i$ and delay $d_i$, and let $b = \{b_1, b_2, \ldots, b_m\}$ denote the bandwidth vector, $d = \{d_1, d_2, \ldots, d_m\}$ denote the delay vector. Then we want to find an $s - t$ path $P$ which satisfies the quality of service requirement, i.e., $\min_{e_i \in P} b_i \geq B$ and $\sum_{e_i \in P} d_i \leq D$, where $B$ and $D$ are given threshold.

The multicast quality of service routing problem with bandwidth and delay is defined as follows: given a graph $(V, E, s, t)$, $V = \{1, 2, \ldots, n\}$, $E = \{e_1, e_2, \ldots, e_m\}$, $s$ is the service provider and $t = \{t_1, t_2, \ldots, t_k\} (k \leq m)$ is the set of terminal users. Let each edge $e_i$ has associated bandwidth $b_i$ and delay $d_i$, and let $b = \{b_1, b_2, \ldots, b_m\}$ denote the bandwidth vector, $d = \{d_1, d_2, \ldots, d_m\}$ denote the delay vector. Then we want to find a tree $T$ covering $s$ and all nodes in $t$ which satisfies the quality of service requirement, i.e., $\min_{e_i \in P_j} b_i \geq B_j$, $\sum_{e_i \in P_j} d_i \leq D_j$, $j = 1, 2, \ldots, k$, where $P_j$ is the unique path from $s$ to $t_j$ on the tree $T$, and $B_j$, $D_j$ are given threshold.

Multicast routing can be viewed as a generalization of unicast routing in many cases. For more details, the readers may refer to the survey paper [4] and papers cited therein.

Conversely, an inverse quality of service routing problem is to modify the parameters as little as possible such that a given $s - t$ path $P$ can satisfy the quality of service requirement for the unicast routing, or a given tree $T$ which covering $s$ and all nodes in $t$ can satisfy the quality of service requirement for the multicast routing. The parameters considered in this paper are bandwidth and delay, and the modification cost is measured by the weighted $l_1$ norm. Note that a path is a special tree, i.e., inverse unicast routing problem is a special case of the inverse multicast routing problem. Hence, we only consider the inverse multicast routing problem in detail.

**Inverse multicast routing problem**
Let each edge $e_i$ has associated bandwidth modification cost $w_i^b$ and delay modification cost $w_i^d$, and let $w^b = \{w_1^b, w_2^b, \ldots, w_m^b\}$ denote the bandwidth modification cost vector, $w^d = \{w_1^d, w_2^d, \ldots, w_m^d\}$ denote the delay modification cost vector. Let $T$ be a given tree which covering $s$ and all nodes in $t$, but under the current $b$ and $d$, the $T$ can not satisfy the quality of service requirement, i.e., $\min_{e_i \in P_j} b_i < B_j$, $\sum_{e_i \in P_j} d_i > D_j$, $j = 1, 2, \ldots, k$. Then for the inverse multicast routing problem with bandwidth and delay under the weighted $l_1$ norm, we look for a new bandwidth vector $b^*$ and a new delay vector $d^*$ such that

(a) the given tree $T$ satisfies the quality of service requirement, i.e., $\min_{e_i \in P_j} b_i^* \geq B_j$, $\sum_{e_i \in P_j} d_i^* \leq D_j$, $j = 1, 2, \ldots, k$;

(b) for each $e_i \in E$, $-l_i^b \leq b_i^* - b_i \leq u_i^b$, $-l_i^d \leq d_i^* - d_i \leq u_i^d$, where $l_i^b, u_i^b, l_i^d, u_i^d \geq 0$ are respectively given bounds for bandwidth and delay;

(c) the total modification cost for change bandwidth and delay of all edges, i.e., $\sum_{e_i \in E} w_i^b |b_i^* - b_i| + \sum_{e_i \in E} w_i^d |d_i^* - d_i|$, is minimized.

In general, in an inverse optimization problem, a feasible solution is given which is not optimal under the current parameter values, and it is required to modify some parameters with minimum modification cost such that the given feasible solution becomes an optimal solution. Burton and Toint [3] were the first who investigate the inverse version of the shortest path problem. Since then, different inverse optimization problems have been well studied when the modification cost is measured by (weighted) $l_1$ norm, $l_2$ norm, $l_\infty$ norm and Hamming distance. Some examples are the inverse minimum cost flow problem [1,5,9], the inverse center location problem [2,12,14], the inverse shortest path problem [3,13,15], the inverse minimum cut problem [7,10,11] and so on. For more details, readers may refer to the survey paper [6] and papers cited therein.

The problem considered in our paper has so far not been treated in literature, but seems to have some potential applications in real world. For example, the service provider has built his own network (path for one to one service, tree for one to many service) to service his customers. At the beginning, the network is optimal for the service provider. But as the service requirement is growing, the exist network is not optimal any more, i.e., it can not satisfies the customers' requirement. Hence, the service provider need to improve the exist network to meet the customers' requirement. Which is the model considered in this paper.

The remainder of the paper is organized as follows. Section 2 considers the inverse multicast quality of service routing problem with bandwidth and delay under the weighted $l_1$ norm. Strongly polynomial combinatorial algorithms are presented. Some final remarks are made in Sect. 3.

## 2 Inverse Multicast Quality of Service Routing Problem

In this section, we are going to consider the inverse multicast quality of service routing problem with bandwidth and delay under the weighted $l_1$ norm, which can be formulated as follows.

$$\begin{aligned} \min & \sum_{e_i \in E} w_i^b |b_i^* - b_i| + \sum_{e_i \in E} w_i^d |d_i^* - d_i| \\ \text{s.t.} & \min_{e_i \in P_j} b_i^* \geq B_j, j = 1, 2, \ldots, k; \\ & \sum_{e_i \in P_j} d_i^* \leq D_j, j = 1, 2, \ldots, k; \\ & -l_i^b \leq b_i^* - b_i \leq u_i^b, i = 1, 2, \ldots, m; \\ & -l_i^d \leq d_i^* - d_i \leq u_i^d, i = 1, 2, \ldots, m. \end{aligned} \tag{1}$$

Note that $b$ and $d$ are independent when they are be changed. Hence problem (1) can be divided into the following two problems:

$$\begin{aligned} \min & \sum_{e_i \in E} w_i^b |b_i^* - b_i| \\ \text{s.t.} & \min_{e_i \in P_j} b_i^* \geq B_j, j = 1, 2, \ldots, k; \\ & -l_i^b \leq b_i^* - b_i \leq u_i^b, i = 1, 2, \ldots, m. \end{aligned} \tag{2}$$

$$\begin{aligned} \min & \sum_{e_i \in E} w_i^d |d_i^* - d_i| \\ \text{s.t.} & \sum_{e_i \in P_j} d_i^* \leq D_j, j = 1, 2, \ldots, k; \\ & -l_i^d \leq d_i^* - d_i \leq u_i^d, i = 1, 2, \ldots, m. \end{aligned} \tag{3}$$

**Theorem 1.** *Suppose $b^*$ is an optimal solution of problem (2) and $d^*$ is an optimal solution of problem (3). Then the optimal solution of problem (1) is $\{b^*, d^*\}$, and the associate optimal value is* $\sum_{e_i \in E} w_i^b |b_i^* - b_i| + \sum_{e_i \in E} w_i^d |d_i^* - d_i|$.

Hence, we need to solve problems (2) and (3), respectively.

For problem (2), it is to modify the bandwidth under the given bound as little as possible such that the given tree $T$ meet the bandwidth constraint. First, it is clear that the bandwidth of the edges out of the tree $T$ need not be changed. Second, for the edges belong to the tree $T$, we need to increase the bandwidth for some of them to meet the bandwidth constraint. Due to the objective function and the bandwidth bound constraint of problem (2), for the edges belong to the unique path from $s$ to $t_j$, if $b_i \geq B_j$, then the bandwidth need not be changed, otherwise, the bound must satisfies $b_i + u_i^b \geq B_j$ and we increasing the bandwidth to $B_j$. Hence, we have the following algorithm to solve problem (2).

**Algorithm 1**

*Step 1.* Let $j = 1$.

*Step 2.* Let $P_j$ denote the unique path from $s$ to $t_j$. If there exist an edge $e_i \in P_j$ such that $b_i + u_i^b < B_j$, then output problem (2) is infeasible and stop.

Otherwise, set

$$u_i^b = \begin{cases} u_i^b, & e_i \notin T, \\ u_i^b, & e_i \in P_j \text{ and } b_i \geq B_j, \\ u_i^b - (B_j - b_i), & e_i \in P_j \text{ and } b_i < B_j, \end{cases}$$

$$b_i = \begin{cases} b_i, & e_i \notin T, \\ b_i, & e_i \in P_j \text{ and } b_i \geq B_j, \\ B_j, & e_i \in P_j \text{ and } b_i < B_j, \end{cases}$$

*Step 3.* If $j = k$, output the current bandwidth vector $b$ as an optimal solution of problem (2). Otherwise, set $j = j + 1$ and go back to the Step 2.

**Theorem 2.** *The Algorithm 1 solves problem (2) with a time complexity* $O(k) \leq O(n)$.

*Proof.* First, from the above analysis, the validity of the Algorithm 1 is straightforward.

Second, let us consider the time complexity of the algorithm. We designate computations starting from Step 2 until switching back to the next Step 2 as one iteration. In each iteration, we will modify the bandwidth of some of the edges on the path $P_j$ to let the associate $t_j$ meet the bandwidth constraint or find problem (2) is infeasible, which means the Algorithm 1 will terminate at most $k$ iterations. Combining with the computation of the Step 2, the Algorithm 1 runs in $O(k) \leq O(n)$. □

For problem (3), it is to modify the delay as little as possible such that the given tree $T$ meet the delay constraint. First, it is clear that the delay of the edges out the tree $T$ need not be changed. Second, for the edges belong to the tree $T$, we need to decrease the delay for some of them to meet the delay constraint. Hence, we can rewrite problem (3) as following.

$$\begin{aligned} &\min \sum_{e_i \in T} w_i^d (d_i - d_i^*) \\ \text{s.t.} &\sum_{e_i \in P_j} d_i^* \le D_j, j = 1, 2, \ldots, k; \\ &0 \le d_i - d_i^* \le l_i^d, i = 1, 2, \ldots, m. \end{aligned} \tag{4}$$

For simplicity, denote $\Delta d_i = d_i - d_i^*$, $\Delta D_j = \sum_{e_i \in P_j} d_i - D_j$. Then problem (4) is simplified to

$$\begin{aligned} &\min \sum_{e_i \in T} w_i^d \Delta d_i \\ \text{s.t.} &\sum_{e_i \in P_j} \Delta d_i \ge \Delta D_j, j = 1, 2, \ldots, k; \\ &0 \le \Delta d_i \le l_i^d, i = 1, 2, \ldots, m. \end{aligned} \tag{5}$$

To solve problem (5), we first consider a special case: $\Delta D_j = C$ for $j = 1, 2, \ldots, k$, where $C$ is a given threshold, i.e., the following problem.

$$\begin{aligned} &\min \sum_{e_i \in T} w_i^d \Delta d_i \\ \text{s.t.} &\sum_{e_i \in P_j} \Delta d_i \ge C, j = 1, 2, \ldots, k; \\ &0 \le \Delta d_i \le l_i^d, i = 1, 2, \ldots, m. \end{aligned} \tag{6}$$

To solve problem (6), we construct a new weighted graph $\widetilde{G}$ based on the given tree $T$. The node set of $\widetilde{G}$ is $V(T) \cup \{r\}$, i.e., add a new node $r$ to the tree $T$. The edge set of $\widetilde{G}$ is $E(T) \cup \{(t_i, r), i = 1, 2, \ldots, k\}$, i.e., besides the tree edges, add new edges connect the terminal nodes $t_i$ and the new node $r$. And we set the weight of the edges as

$$w_i = \begin{cases} w_i^d, & e_i \in T, \\ +\infty, & e_i = (t_i, r). \end{cases}$$

An illustration of the weighted graph $\widetilde{G}$ is shown in Fig. 1.

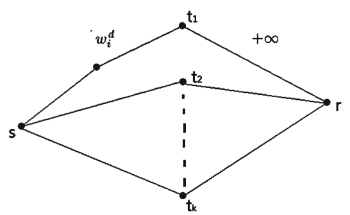


**Fig. 1.** An illustration of the weighted graph $\widetilde{G}$.

**Theorem 3.** *Suppose $M$ is the edge set of a minimum $s-r$ cut of the weighted graph $\widetilde{G}$. Then the following $\{\Delta d_i^*\}$ form an optimal solution of problem (6) with $C \leq \min_{e_i \in M} l_i^d$.*

$$\Delta d_i^* = \begin{cases} C, \; e_i \in M, \\ 0, \;\; e_i \in T \setminus M. \end{cases}$$

*Proof.* First, the $\{\Delta d_i^*\}$ given by the theorem satisfies the first constraint of problem (6) since $M$ is the edge set of a minimum $s-r$ cut of the weighted graph $\widetilde{G}$. Second, the $\{\Delta d_i^*\}$ given by the theorem satisfies the second constraint of problem (6) since we set $C \leq \min_{e_i \in M} l_i^d$ before. Hence, the $\{\Delta d_i^*\}$ given by the theorem is a feasible solution of problem (6).

Furthermore, we say the $\{\Delta d_i^*\}$ given by the theorem is an optimal solution of problem (6). Otherwise, there exist another optimal solution $\{\widetilde{\Delta d_i}\}$. Denote $M' = \{e_i \in T | \widetilde{\Delta d_i} \neq 0\}$. Then for $e_i \in M$, denote $e_i = (x, y)$, $P_x$ is the unique path form $s$ to $x$ and $T_y$ is the subtree whose root is $y$ (we will use the same notation for other nodes in the following). If $\widetilde{\Delta d_i} \neq \Delta d_i^*$ for $e_i \in M$, then at least one of the following two cases will occur.

**Case 1.** $M' \cap T_y \neq \emptyset$. In this case, we can transfer the modification of the delay from $M' \cap T_y$ to $e_i \in M$ without increasing the modification cost since $M$ is a minimum $s-r$ cut of the weighted graph $\widetilde{G}$.

**Case 2.** $M' \cap P_x \neq \emptyset$. For $\widetilde{e_i} = (\widetilde{x}, \widetilde{y}) \in M' \cap P_x$, we can transfer the modification of the delay from $\widetilde{e_i}$ to $M \cap T_y$ without increasing the modification cost since $M$ is a minimum $s-r$ cut of the weighted graph $\widetilde{G}$. Hence, consider the edges belong to $M' \cap P_x$ one by one, we can transfer the modification of the delay from $M' \cap P_x$ to $M$ without increasing the modification cost.

Combining the above two cases, consider the edges belong to $M'$ one by one, we can transfer the modification of the delay from $M'$ to $M$ without increasing the modification cost, which means the $\{\Delta d_i^*\}$ given by the theorem is an optimal solution of problem (6). □

**Theorem 4.** *Suppose $\Delta d_i$ is a feasible solution of problem (5), then $\Delta d_i$ can be presented as $\Delta d_i = \Delta d_i^1 + \Delta d_i^2$, where $\Delta d_i^1$ is a feasible solution of problem (6) and $\Delta d_i^2$ is a feasible solution of the following problem.*

$$\begin{aligned} & \min \sum_{e_i \in T} w_i^d \Delta d_i \\ s.t. & \sum_{e_i \in P_j} \Delta d_i \geq \Delta D_j - C \geq 0, j = 1, 2, \ldots, k; \\ & 0 \leq \Delta d_i \leq l_i^d, i = 1, 2, \ldots, m. \end{aligned} \tag{7}$$

*Proof.* First, let

$$V_h = \{v \in V(T) \mid \text{ the number of the edges of the path from } s \text{ to } v \text{ is } h\}.$$

In this way, $V(T)$ is decomposed to its subsets $V_0, V_1, \ldots, V_q$, where $q$ is the number of the edges of the longest path from $s$ to the nodes in the tree $T$.

Suppose $\varDelta d_i$ is a feasible solution of problem (5), then it is easy to know the $\varDelta d_i^1, \varDelta d_i^2$ given by the following algorithm will satisfy the theorem.

**Algorithm 2**

*Step 1.* Set $h = 0$.

*Step 2.* Select an $e_i = (x, y)$ such that $x \in V_h, y \in V_{h+1}$. Let $\varOmega = C - \sum_{e_i \in P_x} \varDelta d_i^1$, where $P_x$ is the unique path form $s$ to $x$.

*Step 3.* If $\varDelta d_i \leq \varOmega$, then set $\varDelta d_i^1 = \varDelta d_i, \varDelta d_i^2 = \varDelta d_i - \varDelta d_i^1$.
Otherwise, set $\varDelta d_i^1 = \varOmega, \varDelta d_i^2 = \varDelta d_i - \varDelta d_i^1$.

*Step 4.* Set $V_{h+1} = V_{h+1} \setminus \{y\}$. If $V_{h+1} \neq \emptyset$, then go back to the step 2.

*Step 5.* If $h + 1 = q$, stop. Otherwise, set $h = h + 1$ and go back to the step 2. □

**Theorem 5.** *Suppose $\varDelta d_i^{1*}$ and $\varDelta d_i^{2*}$ are optimal solutions of problems (6) and (7), respectively. Then $\varDelta d_i^* = \varDelta d_i^{1*} + \varDelta d_i^{2*}$ is an optimal solution of problem (5).*

*Proof.* First, it is easy to know that $\varDelta d_i^*$ is a feasible solution of problem (5) since $\varDelta d_i^{1*}$ and $\varDelta d_i^{2*}$ are optimal solutions of problems (6) and (7).

Next, we will show that $\varDelta d_i^*$ is an optimal solution of problem (5).

Suppose $\varDelta d_i$ is a feasible solution of problem (5). Then by the Theorem 4, the $\varDelta d_i$ can be represented as $\varDelta d_i^1 + \varDelta d_i^2$ such that $\varDelta d_i^1$ is a feasible solution of problem (6) and $\varDelta d_i^2$ is a feasible solution of problem (7). We have

$$\sum_{e_i \in T} w_i^d \varDelta d_i^1 \geq \sum_{e_i \in T} w_i^d \varDelta d_i^{1*},$$

$$\sum_{e_i \in T} w_i^d \varDelta d_i^2 \geq \sum_{e_i \in T} w_i^d \varDelta d_i^{2*},$$

since $\varDelta d_i^{1*}$ and $\varDelta d_i^{2*}$ are optimal solutions of problems (6) and (7).

Hence

$$\begin{aligned} &\sum_{e_i \in T} w_i^d \varDelta d_i \\ &= \sum_{e_i \in T} w_i^d \varDelta d_i^1 + \sum_{e_i \in T} w_i^d \varDelta d_i^2 \\ &\geq \sum_{e_i \in T} w_i^d \varDelta d_i^{1*} + \sum_{e_i \in T} w_i^d \varDelta d_i^{2*} \\ &= \sum_{e_i \in T} w_i^d \varDelta d_i^*, \end{aligned}$$

which implies $\varDelta d_i^*$ is an optimal solution of problem (5). □

Combining the above analysis, we can solve problem (5) by the following algorithm.

**Algorithm 3**

*Step 1.* Construct a weighted graph $\widetilde{G}$ as showed in the Fig. 1. Let $\Delta d_i = 0$ for $e_i \in T$.

*Step 2.* Find a minimum $s - r$ cut $M$ for the current weighted graph. If the weight of the $s - r$ cut $M$ is equal to $+\infty$, then output problem (5) is infeasible. Otherwise, set

$$\underline{l^d} = \min\{l_i^d \mid e_i \in M \text{ and } l_i^d > 0\},$$

$$\underline{\Delta D} = \min\{\Delta D_j \mid j = 1, 2, \ldots, k \text{ and } \Delta D_j > 0\},$$

$$C = \min\{\underline{l^d}, \underline{\Delta D}\},$$

$$\Delta d_i = \begin{cases} C, & e_i \in M, \\ \Delta d_i, & e_i \in T \setminus M. \end{cases}$$

$$l_i^d = \begin{cases} l_i^d - C, & e_i \in M, \\ l_i^d, & e_i \in T \setminus M. \end{cases}$$

$$w_i = \begin{cases} +\infty, & l_i^d = 0, \\ w_i, & l_i^d > 0. \end{cases}$$

$$\Delta D_j = \Delta D_j - C.$$

*Step 3.* If $\Delta D_j = 0$ for all $j = 1, 2, \ldots, k$, stop and output the current $\{\Delta d_i\}$ as an optimal solution of problem (5). Otherwise, go back to the Step 2.

**Theorem 6.** *The Algorithm 3 solves problem (5) with a time complexity*

$$O(n_1 m_1^2) \leq O(nm^2),$$

*where $n_1$ is the node number of the given tree $T$ and $m_1$ is the edge number of the given tree $T$.*

*Proof.* First, we show the validity of the algorithm.

If the algorithm stops at the Step 2, i.e., there exists at least one $\Delta D_j > 0$ and the weight of the minimum $s-r$ cut of the current weighted graph is equal to $+\infty$, that is to say there exists at least one $s-t_j$ path $P_j$ such that $\sum_{e_i \in P_j} d_i > D_j$ and the delay of the edges on the path $P_j$ can not be changed anymore, which implies problem (5) is infeasible.

We next consider the case that problem (5) is feasible, i.e., the algorithm stops at the Step 3. We designate computations starting from the Step 2 until switching back to the next Step 2 as one iteration. From the Theorem 3, we find an optimal solution of an instance of problem (6) in each iteration. Furthermore, combining all iterations we find an optimal solution of problem (5) due to the Theorem 5.

Finally, we study the time complexity of the Algorithm 3. It is clear that the Step 1 takes $O(m_1)$ to construct the weighted graph $\widetilde{G}$, where $m_1$ is the edge number of the given tree $T$. In each iteration, the main computation is to find a minimum $s - r$ cut which can be done in $O(n_1 m_1)$ [8], where $n_1$ is the node number of the given tree $T$. Furthermore, in each iteration, we will set at least one of $l_i^d$ equals 0 or at least one of $\Delta D_j$ equals 0, which means the algorithm iterates for at most $k + m_1$ times. Hence, the algorithm runs in $O(m_1 + n_1 m_1 \cdot (k + m_1)) = O(n_1 m_1^2) \leq O(nm^2)$ time in the worst case, which is a strongly polynomial time algorithm. □

## 3 Concluding Remarks

In this paper, we study the inverse multicast quality of service routing problem with bandwidth and delay under the weighted $l_1$ norm in detail and present strongly polynomial algorithms.

There are some related inverse problems that deserve further study. First, it is interesting to study the inverse quality of service routing problem with bandwidth and delay with other norms, such as $l_2, l_\infty$ and Hamming distance. Second, it is meaningful to consider the inverse quality of service of routing problem with other parameters. Studying computational complexity results and proposing optimal/approximation algorithms are promising.

## References

1. Ahuja, R.K., Orlin, J.B.: Combinatorial algorithms for inverse network flow problems. Networks **40**, 181–187 (2002)
2. Berman, O., Ingco, D.I., Odoni, A.: Improving the location of minimax facilities through network modification. Networks **24**, 31–41 (1994)
3. Burton, D., Toint, P.L.: On an instance of the inverse shortest paths problem. Math. Program. **53**, 45–61 (1992)
4. Chen, S.G., Nahrsted, K.: An overview of quality of serice routing for next-generation high-speed networks: problems and solutions. IEEE Netw., Special Issue Transm. Distrib. Digit. Video **12**(6), 64–79 (1998)
5. Güler, C., Hamacher, H.W.: Capacity inverse minimum cost flow problem. J. Comb. Optim. **19**, 43–59 (2010)
6. Heuberger, C.: Inverse optimization: a survey on problems, methods, and results. J. Comb. Optim. **8**, 329–361 (2004)
7. Liu, L.C., Yao, E.Y.: A weighted inverse minimum cut problem under the bottleneck type Hamming distance. Asia. Pac. J. Oper. Res. **24**, 725–736 (2007)
8. Orlin, J.B.: Max flows in $O(nm)$ time, or better. In: Proceedings of STOC 2013, pp. 765–774 (2013)
9. Tayyebi, J., Aman, M.: Note on "Inverse minimum cost flow problems under the weighted Hamming distance". Eur. J. Oper. Res. **234**, 916–920 (2014)
10. Yang, C., Zhang, J.Z., Ma, Z.F.: Inverse maximum flow and minimum cut problems. Optimization **40**, 147–170 (1997)
11. Zhang, J.Z., Cai, M.C.: Inverse problem of minimum cuts. Math. Method. Oper. Res. **47**, 51–58 (1998)
12. Zhang, J.Z., Liu, Z.H., Ma, Z.F.: Some reverse location problems. Eur. J. Oper. Res. **124**, 77–88 (2000)

13. Zhang, J.Z., Ma, Z.F., Yang, C.: A column generation method for inverse shortest path problems. ZOR-Math. Method. Oper. Res. **41**, 347–358 (1995)
14. Zhang, B.W., Zhang, J.Z., He, Y.: The center location improvement problem under the Hamming distance. J. Comb. Optim. **9**, 187–198 (2005)
15. Zhang, B.W., Zhang, J.Z., Qi, L.Q.: The shortest path improvement problems under Hamming distance. J. Comb. Optim. **12**, 351–361 (2006)