A Note On Inverse Max Flow Problem Under Chebyshev Norm

Çiğdem Güler and **Horst W. Hamacher**
Department of Mathematics, University of Kaiserslautern
D-67653 Kaiserslautern, Germany
gueler,hamacher@mathematik.uni-kl.de

Abstract

In this paper, we study the inverse maximum flow problem under ℓ_∞-norm and show that this problem can be solved by finding a maximum capacity path on a modified graph. Moreover, we consider an extension of the problem where we minimize the number of perturbations among all the optimal solutions of Chebyshev norm. This bicriteria version of the inverse maximum flow problem can also be solved in strongly polynomial time by finding a minimum $s-t$ cut on the modified graph with a new capacity function.

Keywords: inverse optimization, maximum flows, maximum capacity path, minimum cut

1 Introduction

In the past few decades, optimization problems with estimated problem parameters have drawn considerable attention from researchers. For this kind of problems one often knows a priori an optimal solution based on observations or experiments, but is interested in finding a set of parameters, such that the known solution is optimum and the deviation from the initial estimates is minimized. The problem of recalculating the parameters satisfying the given two conditions is known as *inverse optimization problem*.

Ahuja and Orlin [1] mention, in their paper, that the major application area for inverse optimization is geophysical sciences and it were, indeed, geophysicists to first study such problems. At the beginning of 90's, a well-known study by Burton and Toint [4, 5] attracted the interest of mathematicians to this topic. In their papers, the authors study inverse shortest path problems to predict the movements of earthquakes.

Among several inverse optimization problems inverse combinatorial problems, especially inverse network optimization problems, have been intensely investigated. We refer to Heuberger [14] for a thorough survey on this topic. For network optimization problems the most popular problem parameters to perturb are costs and capacities. Capacity modifications were examined, in particular, for minimum cut and maximum flow problems. Ahuja and Orlin [2] use combinatorial arguments to prove that the inverse minimum cut problem under ℓ_1-norm can be efficiently solved using maximum flow computations in the graph. For Chebyshev norm, the inverse problem requires solving a polynomial sequence of minimum cut problems. Shigeno [20] shows the relationship between the inverse minimum cut problems with lower bounds on arcs under ℓ_∞-norm and the maximum mean-cut problems. Yang *et al.* [21] study inverse minimum cut problems with bound constraints. Moreover, they show that the inverse maximum flow problem is also a maximum flow problem under rectilinear norm. In a recent paper, Zhang and Liu [22] propose strongly polynomial algorithms for the inverse maximum flow problem under the weighted Hamming distance. To the best of our knowledge, there does not exist any studies on the inverse maximum flow problem under ℓ_∞-norm in the literature. In this paper, we close this gap.

Let $G = (N, A)$ be a directed graph with a node set N of n nodes and an arc set A of m arcs. There exist lower and upper flow bounds on the arcs of the digraph, which are denoted by $l : A \to \mathbb{R}^m$ and $u : A \to \mathbb{R}^m$, respectively. In the ***maximum flow problem***, the aim is to find a feasible solution that sends the maximum amount of flow from a specified *source node* s to another specified *sink node* t. It should be noted that the maximum flow problem can be formulated as a minimum cost flow problem by introducing an additional arc (t, s) to the graph G with cost $c_{ts} = -1$ and flow bound $u_{ts} = \infty$ (see Ahuja *et al.* [3]). Hence, the results on inverse minimum cost flows can be carried over to maximum flows. In Güler and Hamacher [11], we showed that the capacity inverse minimum cost flow problem under Chebyshev norm is solvable in $O(nm^2)$ time by a greedy algorithm. Here we prove that the inverse maximum flow problem can, indeed, be solved with an improved time complexity by converting the problem into a maximum capacity path problem. Moreover, we consider an extension of the problem where we minimize the number of perturbations among the optimal solutions of the Chebyshev norm. A similar bicriteria problem was analyzed in Güler and Hamacher [11] for the capacity inverse minimum cost flow problem where among all the optimal solutions of the Chebyshev norm the number of affected arcs was minimized. In Güler and Hamacher [11], we proved that the latter problem is $\mathcal{NP}$-hard. On the other hand, we show in this paper that the bicriteria version of the inverse maximum flow problem can be solved in strongly polynomial time by finding a minimum $s-t$ cut on the modified graph with a new capacity function.

2 Inverse Maximum Flow Problem under ℓ_∞ Norm

Given a nonoptimal feasible flow $\tilde{f} : A \to \mathbb{R}^m$ to an instance of a maximum flow problem on digraph $G = (N, A, l, u)$ and a weight function $w : A \to \mathbb{R}^m_+$, the inverse maximum flow problem under ℓ_∞-norm (denoted subsequently by ℓ_∞-InvMaxFlow) can be formulated as changing the lower and upper bounds such that $\tilde{f}$ will be the maximum flow for the new bounds $\hat{l}$ and $\hat{u}$, and

$$\max_{(i,j)\in A} \max\{w_{ij}|\hat{l}_{ij} - l_{ij}|, w_{ij}|\hat{u}_{ij} - u_{ij}|\}$$

is minimum.

We first review the well-known characterization of the optimality conditions for maximum flows [3]. An *$s-t$ cut* on $G = (N, A)$ is a cut $\omega = (S, \bar{S})$ with $s \in S$ and $t \in \bar{S}$. Let Ω denote the set of all $s-t$ cuts on graph G. We also denote the set of forward arcs of an $s-t$ cut as ω^+, i.e. $(i,j) \in A$ with $i \in S$ and $j \in \bar{S}$, and the set of backward arcs as ω^-, i.e. $(i,j) \in A$ with $i \in \bar{S}$ and $j \in S$. Then, the capacity of an $s-t$ cut is

$$u(\omega) = \sum_{(i,j)\in\omega^+} u_{ij} - \sum_{(i,j)\in\omega^-} l_{ij}.$$

Theorem 1. ***(Max-Flow Min-Cut Theorem)*** *The maximum value of the flow from a source node s to a sink node t in a capacitated network equals the minimum capacity among all $s-t$ cuts.*

By max-flow min-cut theorem (Theorem 1) if a flow f is maximum, then there exists a saturated $s-t$ cut, i.e., there exists a cut ω with $f_{ij} = u_{ij}$ for all $(i,j) \in \omega^+$ and $f_{ij} = l_{ij}$ for all $(i,j) \in \omega^-$. Since in our case $\tilde{f}$ is not a maximum flow, all $s-t$ cuts are unsaturated. That is, for all $s-t$ cuts $\omega \in \Omega$ there exists some $(i,j) \in \omega^+$ with $\tilde{f}_{ij} < u_{ij}$ or $(i,j) \in \omega^-$ with $\tilde{f}_{ij} > l_{ij}$. Consequently, we can reformulate our problem as follows:

Lemma 2. *The inverse maximum flow problem under ℓ_∞-norm is equivalent to finding an $s-t$ cut ω in G such that*

$$c_\omega = \max\{\max_{(i,j)\in\omega^+} w_{ij}(u_{ij} - \tilde{f}_{ij}), \max_{(i,j)\in\omega^-} w_{ij}(\tilde{f}_{ij} - l_{ij})\} \tag{1}$$

is minimum. In particular, it suffices to change the upper bounds for the outgoing arcs of the cut and the lower bounds for the incoming arcs.

In order to solve (1), we define the residual graph $G(\tilde{f}) = (N, A(\tilde{f}))$ with

$$A(\tilde{f}) = (A \backslash \{(i,j) : \tilde{f}_{ij} = u_{ij}\}) \cup \{(j,i) : (i,j) \in A \text{ and } \tilde{f}_{ij} > l_{ij}\}$$

and assign a capacity function $c : A(\tilde{f}) \to \mathbb{R}^{|A(\tilde{f})|}$ with

$$c_{ij} = \begin{cases} w_{ij}(u_{ij} - \tilde{f}_{ij}) & \text{for } (i,j) \in A \\ w_{ij}(\tilde{f}_{ij} - l_{ij}) & \text{for } (i,j) \in A(\tilde{f}) \backslash A. \end{cases} \tag{2}$$

Note that if $\tilde{f}$ is a maximum flow, then there exists an $s-t$ cut $\omega(\tilde{f})$ in $G(\tilde{f})$ such that $\omega(\tilde{f})^+ = \emptyset$.

Lemma 3. *Let $\Omega(\tilde{f})$ denote the set of all $s-t$ cuts in $G(\tilde{f})$. The objective function value of ℓ_∞-InvMaxFlow is equal to*

$$c^* = \min_{\bar{\omega}\in\bar{\Omega}} \max_{(i,j)\in\bar{\omega}^+} c_{ij}. \tag{3}$$

Proof: By the construction of $G(\tilde{f})$, for each $s-t$ cut ω in G there exists an $s-t$ cut $\omega(\tilde{f})$ in $G(\tilde{f})$. Moreover,

$$\omega(\tilde{f})^+ = \{(i,j) : (i,j) \in \omega^+ \text{ with } \tilde{f}_{ij} < u_{ij}\} \cup \{(j,i) : (i,j) \in \omega^- \text{ with } \tilde{f}_{ij} > l_{ij}\}$$

Thus, by Lemma 2, $c^* = \min_{\omega \in \Omega} c_\omega$, which is equal to the objective function of ℓ_∞-InvMaxFlow.

■

Next, we will show that the inverse maximum flow problem under Chebyshev norm can be solved by solving a maximum capacity path problem. The capacity of a directed $s-t$ path P on a graph G is the minimum of the capacities of the arcs in P. Then, the *maximum capacity path problem* (or *bottleneck shortest path problem*) is finding a directed $s-t$ path of maximum capacity [19]. In order to solve the inverse problem as a maximum capacity problem we exploit the bottleneck min-max duality which was first proved by Fulkerson [9] and extended by Edmonds and Fulkerson [7] for clutters, by Hamacher [13] for matroids.

Let E be a finite set. A *family* $\mathfrak{F}$ on E is a family of subsets of E and a *clutter* $\mathfrak{R}$ on E is a family on E such that no member of $\mathfrak{R}$ is contained in another member of $\mathfrak{R}$.

Theorem 4. *For any clutter $\mathfrak{R}$ on a finite set E, there exists a unique clutter $\mathfrak{S} = b(\mathfrak{R})$ on E such that, for any function f from E to $\mathbb{R}$,*

$$\min_{R \in \mathfrak{R}} \max_{x \in R} f(x) = \max_{S \in \mathfrak{S}} \min_{x \in S} f(x). \tag{4}$$

Specifically, $\mathfrak{S}$ is the clutter consisting of the minimal subsets of E that have nonempty intersection with every member of $\mathfrak{R}$.

Any pair of families $\mathfrak{R}$ and $\mathfrak{S}$ on E is called a *blocking system* on E if they satisfy (4) for every f and regardless of whether they are clutters. Edmonds and Fulkerson [7] prove that any blocking system fulfils the following property.

Property 5. *For any partition of E into two sets E_0 and E_1 ($E_0 \cap E_1 = \emptyset$ and $E_0 \cup E_1 = E$), either a member of $\mathfrak{R}$ is contained in E_0 or a member of $\mathfrak{S}$ is contained in E_1, but not both.*

Moreover, Edmonds and Fulkerson [7] show that the $\mathfrak{S} = b(\mathfrak{R})$ specified in Theorem 4 is the one and only clutter on E having the Property 5. Hence, by using Theorem 4 and Property 5 together with the uniqueness of $\mathfrak{S}$ we can derive the following conclusion, which was mentioned by Hamacher [12], as well.

Corollary 6. *Let $G = (N, A)$ be a digraph with $s, t \in N$, and let $c : A \to \mathbb{R}^{|A|}$ be a capacity function. Then,*

$$\max_{P \in \mathcal{P}} \min_{(i,j) \in P} c_{ij} = \min_{\omega \in \Omega} \max_{(i,j) \in \omega^+} c_{ij} \tag{5}$$

where $\mathcal{P}$ is the set of all elementary ***directed*** *$s-t$ paths, Ω is the set of all $s-t$ cuts in G, and ω^+ denote the forward arcs of the cut $\omega \in \Omega$.*

Proof: Let us define $\mathfrak{R}$ to be the set of all elementary directed $s-t$ paths and $\mathfrak{S}$ to be the sets of the forward arcs of all $s-t$ cuts in G. By definition of elementary paths, $s-t$ cuts and clutters, it is obvious that $\mathfrak{R}$ and $\mathfrak{S}$ are clutters. Hence, all we need to show is the validity of Property 5 for $\mathfrak{R}$ and $\mathfrak{S}$.

Consider the capacity function $c : A \to \{0, 1\}$. We define $E_0 = \{(i,j) \in A : c_{ij} = 0\}$ and $E_1 = \{(i,j) \in A : c_{ij} = 1\}$. If the maximum flow from s to t is equal to 1, then there exists an elementary directed path P with $P \subseteq E_1$. By max-flow min-cut theorem (Theorem 1) the minimum capacity $s-t$ cut has a directed arc of capacity 1, which means that there does not exist $\omega^+ \in \mathfrak{S}$ such that $\omega^+ \subseteq E_0$. Similarly if the maximum flow from s to t equals 0, then there exists $\omega^+ \in \mathfrak{S}$ with $\omega^+ \subseteq E_0$ but $\nexists P$ with $P \subseteq E_1$. Hence, the Property 5 holds for $\mathfrak{R}$ and $\mathfrak{S}$, and the results follows from Theorem 4.

■

The main conclusion for the inverse maximum flow problems under ℓ_∞-norm can be derived from Lemma 3 and Corollary 6.

Theorem 7. *The optimum objective function value of ℓ_∞-InvMaxFlow with respect to the nonoptimal flow $\tilde{f}$ on digraph $G = (N, A, l, u)$ can be calculated by solving a maximum capacity (elementary) path problem on the residual graph $G(\tilde{f})$ with respect to the capacities defined by (2).*

The maximum capacity path problem is a well-known combinatorial problem, which has several real-life applications [8, 16]. The problem can be solved in $O(m + n\log n)$ time by modifying Dijkstra's algorithm and using Fibonacci heaps [19]. Gabow [10] employs binary search to solve the problem in $O(m\log_n C)$ time where $C = \|c\|_\infty$ with c being a nonnegative integer capacity vector on arc set. Punnen [17] showed that if a bottleneck combinatorial optimization problem of size m with ordered weights can be solved in $O(\xi(m))$ time, then the problem with arbitrary weights can be solved in $O(\xi(m)\log^*(m))$ time, where $\log^* m$ is the iterated logarithm of m. Thus, the maximum capacity path problem can be solved in $O(m\log^* m)$ time. More recently, Kaibel and Peinhardt [15] proposed an algorithm of $O(m\log\log m)$ running time for the directed graphs with integer arc capacities. For a brief survey of bottleneck network flow problems, one can refer to Punnen and Zhang [18], where a generalized algorithm for the bottleneck network flow problems is provided, as well.

Here we present the Labeling Algorithm, which is a modification of Dijkstra's algorithm. The validity proof of the algorithm follows analogous to the proof of the classical Dijkstra's algorithm.

Algorithm 1. (Labeling Algorithm - Modified Dijkstra's)

1. Set Label$(s) := \infty$ and all other nodes in N to 0. Also assign the set of to be processed and processed nodes with $N^* := \{s\}$ and $N' = \emptyset$.

2. If $N^* = \emptyset$, STOP.
   Else, choose a node $i \in N^*$ and for all outgoing arcs (i, j) assign

$$\text{Label}(j) := \max\{\min\{\text{Label}(i), c_{ij}\}, \text{Label}(j)\}. \tag{6}$$

   If Label$(j) = \min\{\text{Label}(i), c_{ij}\}$, then set Predecessor$(j) := i$.

3. Set $N^* := (N^*\backslash\{i\}) \cup \{j\}$ if $j \notin N'$, and $N' := N' \cup \{i\}$.

Theorem 7 yields, of course, only the optimal objective function value of ℓ_∞-InvMaxFlow. However, once we have the optimum objective function value, we can easily identify an optimum solution. Suppose that c^* is the optimum objective function value, then we set for each arc $(i, j) \in A$,

- $u^*_{ij} = \tilde{f}_{ij}$ if $w_{ij}(u_{ij} - \tilde{f}_{ij}) \le c^*$ and $u^*_{ij} = u_{ij}$ otherwise,
- $l^*_{ij} = \tilde{f}_{ij}$ if $w_{ij}(\tilde{f}_{ij} - l_{ij}) \le c^*$ and $l^*_{ij} = l_{ij}$ otherwise.

It is easy to verify that the pair of lower and upper bound vectors (l^*, u^*) generated in this way is an optimal solution to the inverse maximum flow problem under ℓ_∞-norm.

Note that if we determine an optimum solution in this way, we might have to modify both lower and upper bounds for some arcs. However, by Lemma 2 we know that there exists an optimum solution to the inverse problem where for each arc either the upper bound or the lower bound has to be perturbed. In order to find this solution, we need to find an $s-t$ cut on $\bar{G}$ satisfying Lemma 3. This can be achieved by applying the Minimum Capacity Cut Algorithm (Algorithm 2) of Christofides [6]. This algorithm determines an $s-t$ cut that minimizes the capacity of its maximum capacity arc.

Algorithm 2. (Minimum Capacity Cut Algorithm)

Input: Graph $\bar{G} = (N, \bar{A})$ with capacity $c : \bar{A} \to \mathbb{R}^{|\bar{A}|}$ defined in Lemma 2
Output: An $s-t$ cut $\omega = (S, \bar{S})$ on graph $\bar{G} = (N, \bar{A})$ satifying Lemma 3

1. Start with $s-t$ cut $\bar{K}(\{s\}, N\backslash\{s\})$ on $\bar{G}$ and find the maximum capacity $\bar{c}$ of the forward arcs of $\bar{K}$.
2. Construct the spanning subgraph $G^* = (N, A^*)$ of $\bar{G}$ with $A^* = \{(i, j) \in \bar{A} : c_{ij} \ge \bar{c}\}$.
3. Find the set of reachable nodes $R^*(s)$ from s on the subgraph G^*.
4. If $t \in R^*(s)$, then $c^* = \bar{c}$ and any $s-t$ cut in the spanning subgraph G^* has the maximum capacity c^*. If $t \notin R^*(s)$, go to Step 5.

5. Define $\bar{K}$ as the cut $(R^*(s), N\backslash R^*(s))$ and find the maximum capacity of the arcs in the new cut. Go to Step 2.

In the worst case, the running time of Minimum Capacity Cut Algorithm is $O(mn + m\log m)$, which is slower than the Labeling Algorithm (Algorithm 1) with Fibonacci heaps. Hence, if it is not compulsory to find an optimum $s-t$ cut, it would be more appropriate to use the Labeling Algorithm for solving the inverse problem.

3 Bicriteria Inverse Maximum Flow Problem

An extension of ℓ_∞-InvMaxFlow is a lexicographic bicriteria problem where we minimize the number of perturbations among all the optimum solutions. In this case, the second objective is a unit weight sum-type Hamming distance, i.e.

$$\min \sum_{(i,j)\in A} (H(u_{ij}, \hat{u}_{ij}) + H(l_{ij}, \hat{l}_{ij})) \tag{7}$$

where $H(a, \hat{a}) = 0$ if $\hat{a} = a$ and $H(a, \hat{a}) = 1$ otherwise.

A similar bicriteria problem was analyzed in Güler and Hamacher [11] for the capacity inverse minimum cost flow problem where among all the optimal solutions under Chebyshev norm the number of affected arcs was minimized. There we showed that the bicriteria inverse problem for the minimum cost flows is $\mathcal{NP}$-hard since the capacity inverse minimum cost flow problem under rectilinear (ℓ_1) norm with unit arc capacities is $\mathcal{NP}$-hard. In contrast, Zhang and Liu [22] proved that the inverse maximum flow problem under weighted sum-type Hamming distance is equivalent to solving a minimum $s-t$ cut problem. We propose a similar approach (Algorithm 3) in order to solve the bicriteria inverse maximum flow problem in strongly polynomial time.

Algorithm 3. (Bicriteria Inverse Max Flow Algorithm)

Input: Graph $\bar{G} = (N, \bar{A})$ with capacity $c : \bar{A} \to \mathbb{R}^{|\bar{A}|}$ defined in Lemma 2
Output: An $s-t$ cut $\omega = (S, \bar{S})$ on graph $\bar{G} = (N, \bar{A})$ having the minimum number of forward arcs and satifying Lemma 3

1. Find the optimum objective function value c^* of inverse maximum flow problem under ℓ_∞-norm by solving a maximum path problem on graph $\bar{G}$.
2. Assign a new capacity function $c' : \bar{A} \to \mathbb{R}^{|\bar{A}|}$ for all $(i,j) \in \bar{A}$ such that

$$c'_{ij} = \begin{cases} 1 & \text{if} \quad c_{ij} \le c^* \\ \left(\frac{n^2}{4} + 1\right) & \text{if} \quad c_{ij} > c^* \end{cases} \tag{8}$$

3. Find the minimum $s-t$ cut ω on $\bar{G}$ with the capacity function c'.

Because this algorithm is a slightly modified version of the algorithm in Zhang and Liu [22], we refer to their paper for a correctness proof. The worst case running time of the algorithm is $O(n^3)$ since the most costly operation is identifying the minimum $s-t$ cut in the last step [3].

Once we identify the minimum $s-t$ cut ω on $\bar{G}$, we can generate an optimum solution (l^*, u^*) to the bicriteria inverse maximum flow problem on graph G by assigning

$$l^*_{ij} = \begin{cases} \tilde{f}_{ij} & \text{if} \quad (j,i) \in \omega^+ \\ l_{ij} & \text{otherwise} \end{cases} \qquad u^*_{ij} = \begin{cases} \tilde{f}_{ij} & \text{if} \quad (i,j) \in \omega^+ \\ u_{ij} & \text{otherwise} \end{cases} \tag{9}$$

Acknowledgment

The research has been partially supported by Deutsche Forschungsgemeinschaft (DFG), Grant HA 1737/7 ”Algorithmik großer und komplexer Netzwerke” and by the program ”Center for Mathematical and Computational Modelling (CMCM)”.

References

[1] Ahuja, R. and Orlin, J. (2001). Inverse optimization. *Operations Research*, **49**, 771–783.

[2] Ahuja, R. and Orlin, J. (2002). Combinatorial algorithms of inverse network flow problems. *Networks*, **40**, 181–187.

[3] Ahuja, R., Magnanti, T., and Orlin, J. (1993). *Network Flows: Theory, Algorithms and Applications*. Prentice Hall, New Jersey.

[4] Burton, D. and Toint, P. (1992). On an instance of the inverse shortest paths problem. *Mathematical Programming*, **53**, 45–61.

[5] Burton, D. and Toint, P. (1994). On the use of an inverse shortest paths algorithm for recovering linearly correlated costs. *Mathematical Programming*, **63**, 1–22.

[6] Christofides, N. (1975). *Graph Theory: An Algorithmic Approach*. Academic Press, London.

[7] Edmonds, J. and Fulkerson, D. (1970). Bottleneck extrema. *Journal of Combinatorial Theory*, **8**(2), 299–306.

[8] Fernandez, E., Garfinkel, R., and Arbiol, R. (1998). Mosaicking of aerial photographic maps via seams defined by bottleneck shortest paths. *Operations Research*, **46**(3), 293–304.

[9] Fulkerson, D. (1966). Flow networks and combinatorial operations research. *The American Mathematical Monthly*, **73**, 115–138.

[10] Gabow, H. (1985). Scaling algorithms for network problems. *Journal of Computer and System Sciences*, **31**(2), 148–168.

[11] Güler, Ç. and Hamacher, H. W. (2008). Capacity inverse minimum cost flow problem. *Journal of Combinatorial Optimization*.

[12] Hamacher, H. W. (1976). Verfahren zur Lösung von Flussproblemen mit verallgemeinerten Kosten. Diploma Thesis (in German), Mathematischen Institut der Universität zu Köln. Germany.

[13] Hamacher, H. W. (1981). *Flows in regular matroids*. Ph.D. thesis, Mathematischen Institut der Universität zu Köln, Germany. Published in Mathematical Systems in Economics 69.

[14] Heuberger, C. (2004). Inverse optimization: A survey on problems, methods, and results. *Journal of Combinatorial Optimization*, **8**, 329–361.

[15] Kaibel, V. and Peinhardt, M. (2006). On the bottleneck shortest path problem. *ZIB-Report 06-22*. http://www.zib.de/Publications/abstracts/ZR-06-22/.

[16] Listrovoi, S. and Khrin, V. (1998). Parallel algorithm to find maximum capacity paths. *Cybernetics and Systems Analysis*, **34**(2), 261–268.

[17] Punnen, A. P. (1996). A fast algorithm for a class of bottleneck problems. *Computing*, **56**, 397–401.

[18] Punnen, A. P. and Zhang, R. (2007). Bottleneck flows in networks. *Working Paper*. Online available at http://arxiv.org/abs/0712.3858.

[19] Schrijver, A. (2003). *Combinatorial Optimization - Polyhedra and Efficiency*. Springer Verlag, Berlin.

[20] Shigeno, M. (2002). Minimax inverse problems of minimum cuts. *Networks*, **39**(1), 7–14.

[21] Yang, C., Zhang, J., and Ma, Z. (1997). Inverse maximum flow and minimum cut problems. *Optimization*, **40**, 147–170.

[22] Zhang, J. and Liu, L. (2006). Inverse maximum flow problems under the weighted Hamming distance. *Journal of Combinatorial Optimization*, **12**, 395–408.