Bulletin of the *Transilvania* University of Braşov • Vol 12(61), No. 1 - 2019
Series III: Mathematics, Informatics, Physics, 113-122
https://doi.org/10.31926/but.mif.2019.12.61.1.10

INVERSE MAXIMUM FLOW PROBLEM IN PLANAR NETWORKS

Laura CIUPALĂ[1] and Adrian DEACONU[⋆2]

Dedicated to the 75th birthday of Professor Eleonor Ciurea

Abstract

In this paper we consider the problem of inverse maximum flow in planar network (IMFPN), where upper bounds for the flow must be changed as little as possible so that a given feasible flow becomes a maximum flow in the modified network. A strongly polynomial algorithm for solving this problem is proposed.

2000 *Mathematics Subject Classification:* 90C27, 90C35, 68R10
Key words: inverse combinatorial optimization and maximum flow and planar networks and strongly polynomial time

1 Introduction

The network flow model bridges several diverse and seemingly unrelated areas of combinatorial optimization. More often in scientific writing, flow in a network refers to the flow of electricity, phone calls, e-mail messages, commodities being transported across truck routes, or other such kinds of flow. Many efficient algorithms have been developed to solve the maximum flow problem in networks [1].

The planar network also arises practical contexts such as $V[S]$ design and communication networks, and hence it is of interest to find fast flow algorithms for this class of graphs. The computation of a maximum flow in planar networks has been investigated by many researchers starting from the work of Ford and Fulkerson [10] who developed an $O(n^2)$ time algorithm. Hassin [11] gave an algorithm that runs in $O(nlog(n)^{0.5})$ time using Frederickson's shortest path algorithm. The asymptotically algorithm for maximum flow in planar network is

[1]Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania, e-mail: laura_ciupala@yahoo.com
[2⋆] *Corresponding author*, Faculty of Mathematics and Informatics, *Transilvania* University of Braşov, Romania, e-mail: a.deaconu@unitbv.ro

due to Borradaile and Klein [4]. The maximum flow in planar networks problem has drawn considerable attention from researchers.

In the past few decades, the inverse combinatorial optimization problems have been studied intensively [2][3][12]. For this kind of problems the idea is to modify a vector of parameters (capacities, costs), such that a given feasible solution of the direct optimization problem becomes an optimal solution and the distance between the initial vector and the modified vector of parameters is minimum. Different norms such as l_1 or l_∞ are considered to measure this distance. Capacity modifications are done for maximum flow and minimum cut. Strongly polynomial time algorithms to solve the inverse maximum flow problem (IMF) under l_1 norm were presented by Yang et al. [14]. The inverse maximum flow problem under l_∞ norm is studied by Deaconu [7]. Zhang and Liu [15] propose strongly polynomial algorithms for IMF under Hamming distance. In this paper we study the inverse maximum flow problem in planar networks.

2 Feasible flows and maximum flow

Let $G = (N, A, u)$ be a network with the set nodes $N = \{1, 2, ..., n\}$, the set of arcs $A = \{a_1, a_2, ..., a_n\}$, $a_k = (i, j)$, $i, j \in N$ and the upper bound (capacity) function $u : A \longrightarrow R_+$, where R_+ is the real positive number set. To define the feasible flow problem we distinguish two special nodes in the static network G, a source node 1 and a sink node n. For a given subset $X \subset N$ and for $\bar{X} = N - X$ we use the notation:

$$(X, \bar{X}) = \{(i, j) | (i, j) \in A, i \in X, j \in \bar{X}\} \tag{1}$$

and for the function $g : A \longrightarrow R_+$ we use the notation:

$$g(X, \bar{X}) = \sum_{(X,\bar{X})} g(i, j) \tag{2}$$

A flow is a function $f : A \longrightarrow R_+$ satisfying the nest conditions:

$$f(i, N) - f(N, i) = \begin{cases} v, & \texttt{if } i = 1 \\ 0, & \texttt{if } i \neq 1, n \\ -v, & \texttt{if } i = n \end{cases} \tag{3}$$

where $v \geq 0$. We refer to v as the value of the flow f.

A feasible flow is a flow f which verifies the conditions:

$$0 \leq f(i, j) \leq u(i, j) \tag{4}$$

The maximum flow problem is to determine a feasible flow for which v is maximized.

A cut is a partition of nodes set N into two subsets, X and $\bar{X} = N - X$. We represent this cut using the notation $[X, \bar{X}]$. We refer to a cut $[X, \bar{X}]$ as an $1 - n$ cut if $1 \in X$ and $n \in \bar{X}$. An arc (i, j) with $i \in X$ and $j \in \bar{X}$ is a forward arc of the cut and an arc (i, j) with $i \in \bar{X}$ and $j \in X$ is a backward arc of the cut. Let $(X, \bar{X})$ denote the set of forward arcs and let $(\bar{X}, X)$ denote the set of backward arcs of the cut $[X, \bar{X}]$. For the maximum flow problem in network G the capacity of a $1 - n$ cut $[X, \bar{X}]$ is:

$$c[X, \bar{X}] = c(X, \bar{X}) \tag{5}$$

We refer to a $1 - n$ cut $[X^*, \bar{X}^*]$ whose capacity $c[X^*, \bar{X}^*]$ is the minimum among all $1 - n$ cuts $[X, \bar{X}]$ as a minimum $1 - n$ cut.

Let f be a feasible flow with value v and f^* a maximum feasible flow with value v^*. We recall the next two results [1]:

$$v = f[X, \bar{X}] = f(X, \bar{X}) - f(\bar{X}, X) \leq c[X, \bar{X}] \tag{6}$$

$$v^* = f^*[X^*, \bar{X}^*] = c[X^*, \bar{X}^*] = c(X^*, \bar{X}^*) \tag{7}$$

From 5, 6 and 7 we have:

$$f^*(\bar{X}^*, X^*) = 0, i.e., f^*(i, j) = 0, \forall (i, j) \in (\bar{X}^*, X^*) \tag{8}$$

We consider now that the network G is a directed $1 - n$ planar network. More efficient algorithms can be developed on planar networks. We recall the definitions and properties of planar digraphs [1], [4], [6], [10] and [11].

`Definition 1.` A digraph is a `planar graph` if we can draw it in a two dimensional plane so that no two arcs intersect each other.

Efficient algorithms (in linear time) were developed to test planarity of a digraph.

`Definition 2.` Let $G = (N, A)$ be a planar digraph. A `face` of G is a region of the plane bounded by arcs that satisfies the condition that any two points in the region can be connected by a continuous curve that meets no nodes and arcs. The boundary of a face i' is the set of all arcs that enclose it. The faces i' and j' are called `adjacent` if the intersection of their boundaries contains at least an arc.

Any planar graph has an unbounded face.

Theorem 1. *If a connected graph has n nodes m arcs and q faces then $q = m - n + 2$.*

Theorem 2. *If a planar graph has n nodes and m arcs then $m < 3n$.*

Theorem 3. *A planar digraph remains planar if a multiple arc is added to it.*

Now, we define the dual directed planar network $G' = (N', A', u')$ of the network $G = (N, A, u)$. We add the arc $(n, 1)$ with $u(n, 1) = 0$, which divides the unbounded face into two faces: a new bounded face and a new unbounded face. We place a node i' inside each face i' of the network G. We have $N' = \{1', 2', ..., n'\}$ with $n' = q + 1 = m - n + 3$. Let $1'$ and n' denote, respectively, the nodes in the dual directed network G' corresponding to the new bounded face and the new unbounded face. Each arc $(i, j) \in A$ lies on the boundary of two faces i' and j'. Corresponding to this arc, the network G' contains two opposite arcs (i, j) and (j', i'). If the arc (i, j) is clockwise in the face i' then we define the value $u'(i', j') = u(i, j)$ and the value $u'(j', i') = 0$. We define the arc values in the opposite manner if the arc (i, j) is a counterclockwise arc in the face i'. The network contains the arcs $(1', n')$ and $(n', 1')$ which we delete from the network. We have $A' = \{(i', j'), (j', i') | i', j' \in N', (i', j')$ and (j', i') corresponding to $(i, j) \in A\}$. There is one-to-one correspondence between $1 - n$ cuts in the network G and paths from node $1'$ to n' in the network G'. Moreover, the capacity of each cut equals the value of the corresponding path. Consequently, we can obtain a $1 - n$ cut $[X^*, \bar{X}^*]$ in the network G by determining the shortest path P' from node $1'$ to node n' in G'. We can solve the shortest path problem in G' by using Dijkstra's algorithm [1].

We shall present now an algorithm for finding a maximum flow in a directed $(1, n)$ static planar network $G = (N, A, u)$. Let $d'(i')$ denote the shortest path distance from node $1'$ to i' in the dual directed network G'. We present below the algorithm for maximum flow in a directed $(1, n)$ planar network (algorithm MFDPN).

Algorithm 1 Algorithm MFDPN

```
BEGIN
Compute the network G';
DIJKSTRA(G', d');
for (i, j) ∈ A do
    f*(i, j) = d'(j') − d'(i');
end for
END.
```

Theorem 4. *The algorithm MFDPN determines a maximum flow in the network G.*

Theorem 5. *The algorithm MFDPN determines a maximum flow in $O(n^2)$ time.*

We remark the fact that using Frederickson's algorithm for the shortest path problem, the algorithm MFDPN determines a maximum flow in $O(n^{1.5})$ time.

3 Inverse maximum flow problem in direct planar networks

In the past decades, the inverse maximum flow problem (IMF) has drawn considerable attention from researchers. Strongly polynomial time algorithms to solve this problem were presented [7], [12], [14], [15].

The IMF problem has the following formulation. Let f be a given feasible flow in the network $G = (N, A, u)$. The problem is to change the capacity vector u as little as possible so that f becomes a maximum flow in the modified network.

Throughout the paper we shall consider IMF problem under the l_1 norm:

$$\|u - \hat{u}\|_1 = \sum_{i=1}^{n} |u(a_i) - u(\hat{a}_i)| \tag{9}$$

`Assumption 1:` The network G has a single source node and a single sink node.

`Assumption 2:` The network G is antisymmetric, i.e., if $(i, j) \in A$ then $(j, i) \notin A$.

The above assumptions are not restrictive. It is well known that a network with more than a source node and more than a sink node is equivalent for maximum flow problem to a modified network where a super source node and, respectively, a super sink node are introduced [1]. If G is not antisymmetric then by introducing new nodes in the set N and new arcs in the set A we can build an equivalent antisymmetric network for the maximum flow problem.

The residual capacity for each arc $(i, j) \in A$ is defined as follows:

$$\begin{cases} r(i,j) = u(i,j) - f(i,j) \\ r(j,i) = f(i,j) \end{cases} \tag{10}$$

It is obvious that in order to make a feasible flow f become a maximum flow in the network G the upper bounds (capacities) of some arcs from A must be decreased. So, IMF problem with restrictions on the modification of the capacities can be formulated using the following mathematical model:

$$\texttt{min } \|u - \hat{u}\|_1 \tag{11a}$$

$$f^* = f \texttt{ is a maximum flow in } \hat{G} = (N, A, \hat{u}) \tag{11b}$$

$$u(i, j) - \alpha(i, j) \le \hat{u}(i, j) \le u(i, j),\ \forall (i, j) \in A \tag{11c}$$

where $\alpha(i, j)$ are given positive numbers satisfying the conditions:

$$\alpha(i, j) \le u(i, j), \forall (i, j) \in A \tag{12}$$

We shall present now an algorithm from [14] adapted for a planar network. For all $1 - n$ cuts $[X, \bar{X}]$ with $f(\bar{X}, X) = 0$ we define the vector u' as follows:

$$u'(i, j) = \begin{cases} f(i,j), (i,j) \in (X, \bar{X}) \\ f(i,j), (i,j) \in A - (X, \bar{X}) \end{cases} \tag{13}$$

We denote by C' the following set of $1 - n$ cuts:

$$C' = \{[X, \bar{X}] | f(\bar{X}, X) = 0, u(X, \bar{X}) - u'(X, \bar{X}) \le \alpha(X, \bar{X})\} \tag{14}$$

Theorem 6. *IMF problem has solution for a given feasible flow f if and only if $C' \neq \phi$.*

Proof. If IMF problem has a solution u^* then u^* satisfies the relations from 11. Since $f^* = f$ is a maximum flow in $G^* = (N, A, u^*)$, we have $f^*(X^*, \bar{X}^*) = u^*(X^*, \bar{X}^*)$ and $f^*(\bar{X}^*, X^*) = 0$. From the fact that $u(X^*, \bar{X}^*) - u^*(X^*, \bar{X}^*) \leq \alpha(X^*, \bar{X}^*)$ and $f^*(\bar{X}^*, X^*) = f(\bar{X}^*, X^*) = 0$ it results that $[X^*, \bar{X}^*] \in C'$ and, so, $C' \neq \phi$.

Conversely, if $C' \neq \phi$ then u' defined in 13 is an optimal solution of IMF problem 11, because $u'(X, \bar{X}) = f(X, \bar{X})$ and $f(X, \bar{X}) = 0$, where $[X, \bar{X}] \in C'$.

It is obviously that:

$$u^* \leq u \tag{15}$$

If $[X^*, \bar{X}^*]$ is a minimum $1 - n$ cut in the network $G^* = (N, A, u^*)$, where u^* is the optimal solution of IMF problem 11 then from 6, 7, 8, 11, 13 and 15 we have:

$$[X^*, \bar{X}^*] \in C', u'[X^*, \bar{X}^*] = u^*[X^*, \bar{X}^*] \tag{16}$$

and

$$\|u - u^*\|_1 = min\{u[X^*, \bar{X}^*] | [X^*, \bar{X}^*] \in C'\} - v \tag{17}$$

where v is the value of the feasible flow f.

Let $[X^*, \bar{X}^*] \in C'$ be a $1 - n$ cut with the property:

$$u[X', \bar{X}'] = min\{u[X, \bar{X}] | [X, \bar{X}] \in C'\} \tag{18}$$

We have: $u'(X', \bar{X}') = f(X', \bar{X}')$ and $f(\bar{X}', X') = 0$. Consequently, u' is a feasible solution of the IMF problem and a maximum flow in the network $G' = (N, A, u')$. It results that $[X', \bar{X}']$ is a minimum $1 - n$ cut in G'. We also have that $\|u - u'\|_1 = u(X', \bar{X}') - v$ and, comparing this result with 17 we obtain that $u' = u^*$, i.e., u' is an optimal solution of IMF problem 11. Consequently, IMF problem can be transformed into the following problem:

$$min\{u[X, \bar{X}] | [X, \bar{X}] \in C'\} \tag{19}$$

In order to eliminate the constraint $[X, \bar{X}] \in C'$ we construct an extended network $G_1 = (N_1, A_1, u_1)$, where $N_1 = N$, $A_1 = A \cup A_2$, $u_1 : A_1 \longrightarrow R_+$ with $A_2 = \{(j, i) | (i, j) \in A, f(i, j) > 0\}$ and for each arc $(i, j) \in A$ we define:

$$u_1(i, j) = \begin{cases} u(i, j), r(i, j) \leq \alpha(i, j) \\ \bar{u}, otherwise \end{cases} \tag{20}$$

while for each arc (j, i) we have:

$$u_1(j, i) = \bar{u}, \tag{21}$$

where $\bar{u} = u(a_1) + u(a_2) + ... + u(a_n) + 1$

□

Let $[X_1^*, \bar{X}_1^*]$ be a minimum $1 - n$ cut in the network G_1 with the capacity $c_1[X_1^*, \bar{X}_1^*]$.

Theorem 7. *(1) If $c_1[X_1^*, \bar{X}_1^*] < \bar{u}$ then $[X^*, \bar{X}^*] = [X_1^*, \bar{X}_1^*] \bigcap A$ is a solution of the problem 19.*

(2) If $c_1[X_1^*, \bar{X}_1^*] \geq \bar{u}$ then $C' = \phi$ and the IMF problem has no feasible solution.

Proof. (1) Since $N_1 = N$ we have $X_1^* = X^*$ and $\bar{X}_1^* = \bar{X}^*$. If $c[X_1^*, \bar{X}_1^*] < \bar{u}$ then $f(X^*, \bar{X}^*) = 0$, otherwise there is an arc $(j, i) \in (\bar{X}^*, X^*) \subset A$ with $f(j, i) > 0$ and therefore there is an arc $(i, j) \in (X_1^*, \bar{X}_1^*)$ with $u_1(i, j) = \bar{u}$ and thus $c[X_1^*, \bar{X}_1^*] \geq \bar{u}$ which is a contradiction. We also have that $u(X^*, \bar{X}^*) - u'(X^*, \bar{X}^*) \leq \alpha(X^*, \bar{X}^*)$, otherwise there is an arc $(i, j) \in (X^*, \bar{X}^*)$ with $u_1(i, j) = \bar{u}$ which is again a contradiction. These results imply that $[X^*, \bar{X}^*] \in C'$. We will show now that for each $[X, \bar{X}] \in C'$ we have $c_1(X, \bar{X}) = u_1(X, \bar{X}) = u(X, \bar{X}) = c(X, \bar{X})$. Consequently, the $1 - n$ cut $[X^*, \bar{X}^*]$ is a solution of the problem 19.

(2) If $C' \neq \phi$ there would exist $[X, \bar{X}] \in C'$ and we showed in 1 that each arc $(i, j) \in (X, \bar{X}) \cap A_1$ has the capacity $c_1(i, j) = u(i, j)$ and hence $c_1(X, \bar{X}) < \bar{u}$, which is a contradiction. Therefore, $C' = \phi$ and from theorem 6 it results that IMF problem has no feasible solution. □

We consider next that the network $G = (N, A, u)$ is a directed $1 - n$ planar network.

Theorem 8. *The network $G_1 = (N_1 = N, A1, u_1)$ is a planar network.*

Proof. This theorem is a direct consequence of theorem 3. □

Now we are ready to state the algorithm for solving IMF problem in a directed $1 - n$ planar network G.

Algorithm 2 Algorithm IMFPPN

BEGIN
Compute the network $G_1 = (N_1, A_1, u_1)$;
Compute the dual network $G_1' = (N_1', A_1', u_1')$;
DIJKSTRA(G_1', d_1');
if $d_1'(n_1') < \bar{u}$ **then**
  if $(i, j) \in (X^*, \bar{X}^*)$ **then** $u^*(i, j) = f(i, j)$
  else $u^*(i, j) = u(i, j)$
  end if
else IMF does not have feasible solution
end if
END.

Theorem 9. *Algorithm IMFPPN solves the IMF problem in a directed $1 - n$ planar network $G = (N, A, u)$ in $O(n^2)$ time.*

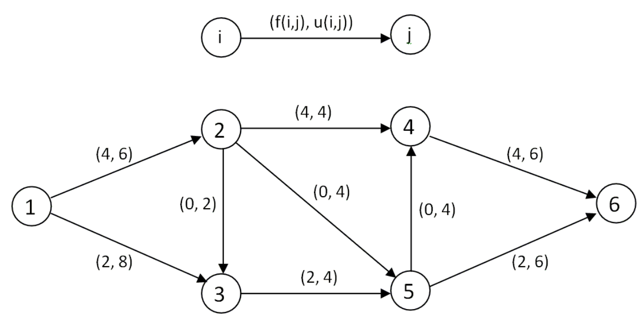

Figure 1: The $1-n$ planar network $G=(N,A,u)$

Proof. We have $|N_1| = |N| = n$, $|A| = m$, $O(m) = O(n)$, $|A_1| = m_1$, $m_1 < 2m$, $|N_1'| = n_1' = q + 1 = m_1 - n_1 + 3 \Rightarrow O(m_1') = O(m_1) = O(m) = O(n)$. Dijkstra algorithm has a time complexity of $O((n_1')^2) = O(n^2)$. Hence, the algorithm IMFPPN has a time complexity of $O(n^2)$. □

We remark the fact that using Frederickson's algorithm for the shortest path problem, the algorithm IMFPPN solves IMF problem in $O(n^{1.5})$ time.

We shall take now a numerical example to illustrate how the algorithm IMFPPN works.

We consider the $1-n$ planar network $G=(N,A,u)$ from figure 1, where $\alpha(i,j) = 2$ for all arcs (i,j) from A.

The planar network $G_1 = (N_1, A_1, u_1)$ and the dual network $G_1' = (N_1', A_1', u_1')$ are shown in figure 2. The values of the arcs $(i', j') \in A'$ are presented in matrix V:

$$v(i,j) = \begin{cases} u_1(i,j), (i',j') \in A_1' \\ w, otherwise \end{cases} \tag{22}$$

where w is a number with the property that $w > \bar{u}$.

$$V = \begin{bmatrix}
w & \bar{u} & w & w & 4 & w & w & w & \bar{u} & w & w & w \\
0 & w & 0 & w & w & w & w & w & w & w & w & w \\
0 & \bar{u} & w & 0 & w & 2 & w & w & w & w & w & w \\
w & w & \bar{u} & w & w & w & w & w & w & w & w & 6 \\
0 & w & w & w & w & 0 & w & 0 & w & 0 & w & w \\
w & w & 0 & w & \bar{u} & w & \bar{u} & w & w & w & w & w \\
w & w & w & w & w & 0 & w & w & w & w & w & w \\
w & w & w & w & w & w & \bar{u} & w & w & 0 & w & 4 \\
0 & w & w & w & w & w & w & w & w & w & w & w \\
w & w & w & w & w & w & \bar{u} & w & \bar{u} & w & 0 & w \\
w & w & w & w & w & w & w & w & w & \bar{u} & w & 6 \\
w & w & w & 0 & w & w & w & 0 & w & w & 0 & w
\end{bmatrix}$$

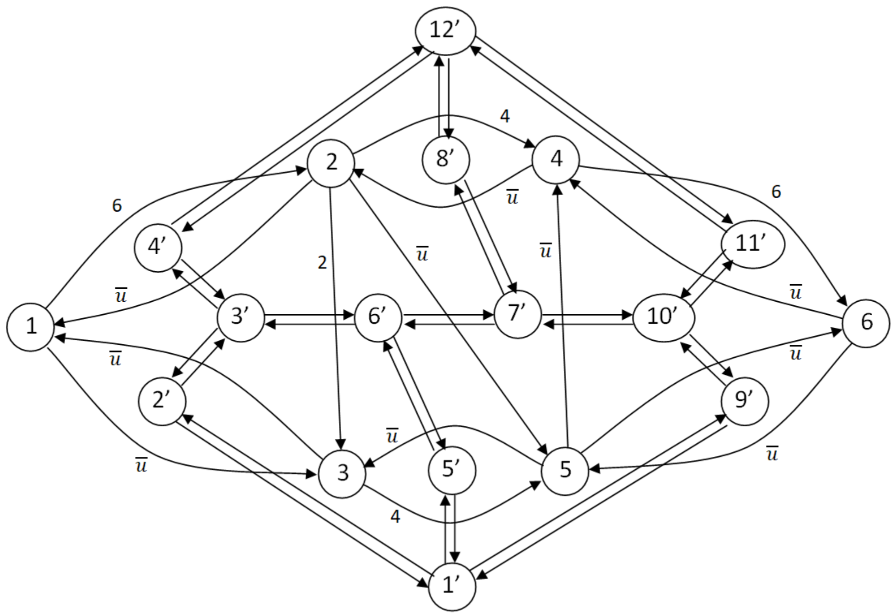

Figure 2: The planar network G_1 and the dual network G_1'

We obtain: $d_1' = (0, \bar{u}+4, 4, 4, 4, 4, \bar{u}+4, \bar{u}+4, \bar{u}, \bar{u}, \bar{u}, 10)$, $X^* = \{1, 2, 3\}$, $u^* = (u^*(1,2), u^*(1,3), u^*(2,3), u^*(2,4), u^*(2,5), u^*(3,5), u^*(4,6), u^*(5,4), u^*(5,6)) =$
$= (4, 8, 0, 4, 4, 2, 6, 4, 6)$.

References

[1] Ahuja, R. K., Magnanti, T. L., Orlin, J. B., *Network flows: theory, algorithms, and applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] Ahuja, R. K., Orlin, J. B., *Combinatorial algorithms for inverse network flow problems*, Networks, (2002).

[3] Ahuja, R. K., Orlin, J. B., *Inverse optimization*, Working Paper, Sloan School of Management, MIT, Cambridge, MA, 1998

[4] Borradaile, G., Klein, P., *An O(n log(n)) algorithm for maximum flow st-flow in a directed planar graph*, ACM Journal **56** (2009), no 2, 1-30.

[5] Ciurea, E. and Deaconu, A., *Inverse minimum flow problem*, Journal of Applied Mathematics and Computing **23** (2007), 193-203.

[6] Ciurea, E. and Georgescu, O., *Minimum flows in directed s-t planar networks*, Bul. Math. de la Soc. des Scien. Math. de Roumanie **101** (2007), no. 2, 305-313.

[7] Deaconu, A., *The inverse maximum flow problem considering L_∞ norm*, RAIRO-Oper. Res. **42** (2008), no. 3, 401-414.

[8] Deaconu, A., *Solving inverse maximum flow problem using weakly polynomial methods*, Bull. Transilvania University Braşov **14(49)** (2007), 83-90.

[9] Demange, M. and Monnot, J. *An introduction to inverse combinatorial problems*, In: Vangelis Th. Paschos, Paradigms of Combinatorial Optimization (Problems and New approaches), Wiley, London-Hoboken 2010.

[10] Ford, L. and Fulkerson, D., *Flows in networks*, Princeton University Press, 1962.

[11] Hassin, R., *Maximum flow in (s,t) planar networks*, Information Processing Latters **13**(3) (1981) 107.

[12] Heuberger,C., *Inverse combinatorial optimization, a survey on problems, methods, and results*, Journal of Combinatorial Optimization, **8** (2004), 329-361.

[13] Tayyebi, J., Mohammadi, A. and Kazemi, S. M. R., *Reverse maximum flow problem under the weighted Chebyshev distance*, accepted for publication, RAIRO-Oper. Res. (2018).

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

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