RAIRO-Oper. Res. 42 (2008) 401–414
DOI: 10.1051/ro:2008017

RAIRO Operations Research
www.rairo-ro.org

THE INVERSE MAXIMUM FLOW PROBLEM CONSIDERING l_∞ NORM

Adrian Deaconu[1]

Abstract. The problem is to modify the capacities of the arcs from a network so that a given feasible flow becomes a maximum flow and the maximum change of the capacities on arcs is minimum. A very fast $O(m \cdot \log(n))$ time complexity algorithm for solving this problem is presented, where m is the number of arcs and n is the number of nodes of the network. The case when both, lower and upper bounds of the flow can be modified so that the given feasible flow becomes a maximum flow is also discussed. The algorithm proposed can be adapted to solve this problem, too. The inverse minimum flow problem considering l_∞ norm is also studied.

Keywords. Inverse combinatorial optimization, maximum flow, strongly polynomial time complexity.

Mathematics Subject Classification. 90C27, 90C35, 68R10.

1. Introduction

An inverse combinatorial optimization problem consists in modifying some parameters of a network such as capacities or costs so 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, l_∞ and even l_2 are considered to measure this distance. In the last years many papers were published in the field of inverse combinatorial optimization [8]. Almost every inverse problem was studied considering l_1 and l_∞ norms, resulting in different problems with completely different solution methods. Strongly polynomial time algorithms to solve the inverse maximum flow problem

Received September 07, 2006. Accepted March 28, 2008.

[1] University "Transilvania" of Brasov, Faculty of Mathematics and Informatics, Theoretical Computer Science Departement, str. Iuliu Maniu 50, Brasov, Romania; a.deaconu@unitbv.ro

Article published by EDP Sciences

when l_1 norm is considered (denoted IMF) were presented by Yang *et al.* [12]. IMF is reduced to a minimum cut problem in an auxiliary network with finite and infinite arc capacities. The algorithm for IMF has an $O(n \cdot m \cdot \log(n^2/m))$ time complexity.

The more general case (denoted GIMF) is studied in [6], where lower and upper bounds for the flow are changed. Strongly and weakly polynomial algorithms to solve GIMF are proposed. The strongly polynomial algorithms for GIMF have the same time complexity as the algorithms for IMF, but the minimum cut is searched in a network with fewer arcs. The weakly polynomial algorithms for GIMF have an $O(\min\{n^{2/3}, m^{1/2}\} \cdot m \cdot \log(n^2/m) \cdot \log(\max\{n, R\}))$ time complexity, where $R = \max\{c(x,y) - f(x,y) + f(y,x) - l(y,x) | x, y \in N\}$.

The least number of modifications to the lower or/and upper bounds is considered in [7]. An $O(\min\{n^{2/3}, m^{1/2}\} \cdot m)$ algorithm is proposed to solve this problem.

Four inverse maximum flow problems are also studied by Liu and Zhang [9] under the sum-type and bottleneck-type weighted Hamming distance. Strongly polynomial algorithms to solve these problems are proposed.

The (direct) problems of the maximum flow problem and the minimum cut problem are connected by strong duality. In fact, the minimum cut can be obtained after the maximum flow is found. The inverse minimum cut problem (denoted IMC) consists in modifying as little as possible the bounds for the flow so that a given cut becomes a minimum cut. IMC can be reduced to an inverse maximum flow problem under l_1 norm and the more general case with bound constrains can be reduced to a maximum-weight flow problem [12]. When solving IMC, the capacities are changed only on the arcs of the given cut, and therefore the optimal solution to IMC may not be optimal for IMF (the inverse problems do not share the strong duality property of their direct counterpart).

In this paper, the inverse maximum flow considering l_∞ norm (denoted IMF∞) is studied. While in the case of IMF the sum of changes brought to the capacities of the arcs is minimized, in the case of IMF∞ the maximum change is minimized. Depending on the situation, one of the two ways of minimizing the change to the capacities of the arcs can be considered. Most of the inverse problems are reduced to their direct problems or to related direct optimization problems in special networks. From this point of view IMF∞ can be considered atypical.

When the total amount of change must be minimum, algorithms for the IMF problem can be applied and if the maximum change to the capacities on arcs is needed to be minimum, the algorithm for IMF∞ can be applied. One of the advantages of considering the algorithm for IMF∞ is that it is very fast, it has a time complexity of $O(m \cdot \log(n))$, where m is the number of arcs and n is the number of nodes of the network. The algorithms for IMF problem run considerably slower, because they depend on the time complexity of the method used to find the minimum cut, which is $O(n \cdot m \cdot \log(n^2/m))$ in the strongly polynomial case.

In Section 2 $IMF\infty$ is introduced. A method to test in linear time if the inverse maximum flow or the inverse minimum flow problems have optimal solution is also presented in Section 2. This method does not depend on the chosen norm.

In Section 3 an algorithm to solve $IMF\infty$ is presented.

In Section 4 a numerical example is given.

The more general case when, both the upper and lower bounds for the flow are considered for modification is studied in Section 5. In this section the algorithm for IMF∞ is adapted to solve the inverse minimum flow problem.

2. The IMF∞ problem

Let $G = (N, A, c, s, t)$ be an s-t network, where N is the set of nodes, A is the set of directed arcs, c is the capacity vector, s is the source and t is the sink node.

If a network has more than one source or/and more than one sink node, it can be transformed into an s-t network (introducing a super-source and a super-sink node) [1].

Let f be a given feasible flow in the network G. It means that f has to satisfy the flow balance condition and the capacity restrictions. The balance condition for the flow f is:

$$\sum_{y\in N,(x,y)\in A} f(x,y) - \sum_{y\in N,(y,x)\in A} f(y,x) = \begin{cases} v(f), & x = s \\ -v(f), & x = t \\ 0, & x \in N - \{s,t\}, \end{cases} \quad ,\forall\, x \in N \tag{1}$$

where *v(f)* is the value of the flow f from s to t.

The capacity restrictions are:

$$0 \le f(x,y) \le c(x,y),\ \forall\ (x,y) \in A. \tag{2}$$

The maximum flow problem is:

$$\begin{cases} \texttt{max}\ v(f) \\ f\ \texttt{is a feasible flow in}\ G. \end{cases} \tag{3}$$

The residual network attached to the network G for the flow f is $G_f = (N, A_f, r, s, t)$, where for each pair of nodes (x, y) the value of $r(x, y)$ is defined as follows:

$$r(x,y) = \begin{cases} c(x,y) - f(x,y) + f(y,x), & \texttt{if}\ (x,y) \in A\ \texttt{and}\ (y,x) \in A \\ c(x,y) - f(x,y), & \texttt{if}\ (x,y) \in A\ \texttt{and}\ (y,x) \notin A \\ f(y,x), & \texttt{if}\ (x,y) \notin A\ \texttt{and}\ (y,x) \in A \\ 0, & \texttt{otherwise}. \end{cases} \tag{4}$$

The set A_f contains as arcs of the residual network only the pairs of nodes $(x, y) \in N \times N$ for which the residual capacity is positive, *i.e.*, $r(x, y) > 0$.

The inverse maximum flow problem considering l_∞ norm is to change the capacity vector c so that the given feasible flow f becomes a maximum flow in G and the maximum change on arc capacities is minimum.

IMF∞ can be formulated using the following mathematical model:

$$\begin{cases} \texttt{min } \|c-\bar{c}\|_\infty \\ f \texttt{ is a maximum flow in } \bar{G}=\{N,A,\bar{c},s,t\} \\ c(x,y)-\delta(x,y)\le\bar{c}(x,y)\le c(x,y)+\alpha(x,y),\ \forall\,(x,y)\in A, \end{cases} \tag{5}$$

where $\alpha(x,y)$ and $\delta(x,y)$ are given non-negative numbers and $\delta(x,y)\le c(x,y)$, for each arc $(x,y)\in A$. These values show how much the capacities of the arcs can vary.

In order to make the flow f a maximum flow in the network G, the capacities of some arcs from A must be decreased. So, the conditions $\bar{c}(x,y)\le c(x,y)+\alpha(x,y)$, for each arc $(x,y)\in A$ have no effect and, instead of (5), the following mathematical model is considered:

$$\begin{cases} \texttt{min } \|c-\bar{c}\|_\infty \\ f \texttt{ is a maximum flow in } \bar{G}=\{N,A,\bar{c},s,t\} \\ c(x,y)-\delta(x,y)\le\bar{c}(x,y),\ \forall\,(x,y)\in A. \end{cases} \tag{5'}$$

When solving IMF∞, if the capacity is changed on an arc (x,y), then it will be decreased with the amount of $c(x,y)-f(x,y)$. If not so, then there still is an augmenting path from s to t that contains the direct arc (x,y) and the modification of the capacity is useless. This means that if $c(x,y)>f(x,y)+\delta(x,y)$ on an arc (x,y), then, when solving IMF∞, there is no need to change the capacity on (x,y).

Let's determine the arcs in the network G on which the capacity will not be changed.

First, as it has been seen, changing the capacity has no effect on an arc (x,y) with $c(x,y)>f(x,y)+\delta(x,y)$. So, it is no need to try changing the capacities of the arcs from the following set:

$$\widetilde{A}_1=\{(x,y)\in A\,|\,f(x,y)+\delta(x,y)<c(x,y)\,\}. \tag{6}$$

Let us suppose that in the network G there are two arcs (x,y) and (y,x) so that $c(x,y)>f(x,y)$ and $f(y,x)>0$. The change of the capacity on the arc (x,y) is useless. The flow f can not be stopped from being increased on a path from s to t that contains the direct arc (x,y) if the capacity of (x,y) is changed, because in the path, instead of the arc (x,y), the arc (y,x) can be considered, which has $f(y,x)>0$. When solving IMF∞, changing the capacity of the arcs from the following set is useless and can lead to a capacity vector which is not an optimal solution for the problem (5′):

$$\widetilde{A}_2=\{(x,y)\in A\,|\,(y,x)\in A \texttt{ and } f(y,x)>0\,\}. \tag{7}$$

It is easy to see that if there is a path from s to t in the network G that contains only direct arcs (x,y) so that $c(x,y)>f(x,y)+\delta(x,y)$ and/or inverse arcs (y,x) with $f(y,x)>0$, then IMF∞ has no solution.

A graph denoted $\widetilde{G} = (N, \widetilde{A})$ can be constructed to verify the feasibility of the solution, where:

$$\widetilde{A} = \widetilde{A}_1 \cup \{(x, y) \in N \times N \,|(y, x) \in A \ \texttt{and} \ f(y, x) > 0\}. \tag{8}$$

Let's observe that $\widetilde{A}_1 \cup \widetilde{A}_2 \subseteq \widetilde{A} \subseteq A_f$.

We have the following theorem:

Theorem 2.1 (existence of an optimal solution)**.** *In the network G, IMF∞ has optimal solution for the given flow f, if and only if there is no directed path in the graph $\widetilde{G}$ from the node s to the node t.*

Proof. Let $G' = (N, A, c')$ be a network for which the last conditions from $(5')$ hold: $c(x, y) - \delta(x, y) \leq c'(x, y), \forall (x, y) \in A$. Let $G'_f = (N, A'_f, r')$ be the residual network attached to the network G' for the flow f. It is easy to see that $r'(x, y) > 0, \forall (x, y) \in \widetilde{A}$ due to the restriction on the capacity vector for the arc (x, y) of G $(c(x, y) > f(x, y) + \delta(x, y) \Rightarrow c'(x, y) > f(x, y))$ or because $(y, x) \in A$ and $f(y, x) > 0$. This means that $\widetilde{A} \subseteq A'_f$.

If IMF∞ is a feasible problem, then it means that there is a vector $\bar{c}$ with $c(x, y) - \delta(x, y) \leq \bar{c}(x, y), \forall (x, y) \in A$ and for which the flow f is a maximum flow in the network $\bar{G} = (N, A, \bar{c})$. Since $\widetilde{A} \subseteq \bar{A}_f$ (from the observation above), if a directed path exists in $\widetilde{G}$, it corresponds to a directed path in $\bar{G}_f$, which leads to an augmentation to the flow f in G (contradiction).

Now, for the inverse implication we construct the following capacity vector for the arcs of the network G:

$$c''(x, y) = \begin{cases} c(x, y), & \texttt{if} \ c(x, y) > f(x, y) + \delta(x, y) \\ f(x, y), & \texttt{otherwise}. \end{cases}$$

It is easy to see that $c(x, y) - \delta(x, y) \leq c''(x, y), \forall (x, y) \in A$. In the residual network $G''_f = (N, A''_f, r'')$ attached to $G'' = (N, A, c'')$ and to the flow f we have $r''(x, y) = 0$, for all $(x, y) \in (N \times N) - \widetilde{A}$. Since $\widetilde{A} \subseteq A''_f$, it means that $\widetilde{A} = A''_f$ (see (4)). Therefore, because there is no path from s to t in the graph $\widetilde{G}$, it results that there is no directed path from s to t in G''_f. This implies that the flow f is a maximum flow in the network $G'' = (N, A, c'')$. It means that c'' is a feasible solution for IMF∞. In IMF∞, the feasible region for the capacity vector $\bar{c}$ can be reduced to $c - \delta \leq \bar{c} \leq c$ (from $(5')$ and because when solving IMF∞ there is no need to increase the values of the capacities), which is a compact region. So, because IMF∞ has a feasible solution, it results that IMF∞ has optimal solution. □

The verification of IMF∞ being feasible can be done in $O(p)$ time complexity, using a graph search algorithm in $\widetilde{G}$, where p is the number of arcs in the set $\widetilde{A}$ with $p \leq m$. Moreover, this test of feasibility can be applied to any inverse maximum flow problem.

Transforming the flow f into a maximum flow in the network G is equivalent to eliminating arcs of the residual network G_f so that the source node s will no longer communicate with the sink node t (there is no directed path in G_f from s to t). Setting the residual capacity on these arcs to 0 does the elimination of arcs. At this point, a very important observation we shall give for the algorithm is that the arcs from $\widetilde{A}$ in the residual network will not be eliminated. If IMF∞ is feasible (see Th. 2.1) and all the arcs in the set $A_f - \widetilde{A}$ are eliminated, then the flow f becomes a maximum flow in the network G with the modified capacity vector. The arcs from $A_f - \widetilde{A}$ must be determined so that if they are eliminated from A_f, the flow f becomes a maximum flow in G and the corresponding modified capacity vector c^* is an optimal solution for the problem $(5')$, *i.e.*, $\|c - c^*\|_\infty$ is minimum.

The arcs from $A_f - \widetilde{A}$ with the lowest capacities will be eliminated from the residual network G_f, because in $(5')$ l_∞ norm is considered and only the biggest change matters.

3. Algorithm for IMF∞

The algorithm for IMF∞ starts with a set $H = A_f - \widetilde{A}$. If IMF∞ is feasible, then in the residual network the elimination of all arcs from H transforms the flow f into a maximum flow in the network G. The arcs of H are sorted descending by their residual capacities. They are eliminated sequentially from H (from the greatest capacity to the lowest) till the eliminated arcs and the arcs from $\widetilde{A}$ form a graph in which there is a directed path from s to t. The remaining arcs in the set H are the arcs that will be eliminated from the residual network G_f. The flow f will become a maximum flow in G, after the modification is done to the capacities.

The algorithm for IMF∞ is:

```
Construct the graph G̃ = (N, Ã);
B := Ã;
Find the set W of the nodes accessible from the source node s
 in the graph G' = (N, B), eliminating the visited arcs of B;
If t ∈ W then
   The IMF∞ problem has no solution;
   STOP.
endif;
Construct the residual network G_f = (N, A_f);
H := A_f − Ã;
Sort descending by the residual capacity the arcs from the set H;
For (x, y) ∈ H do
   If y ∉ W then
      B := B ∪ {(x, y)};
      If x ∈ W then
         Find the nodes W' accessible from the nodes of W in the
          graph G' = (N, B), eliminating the visited arcs of B;
         W := W ∪ W';
```

```
      endif;
   endif;
   If t ∈ W then
      d := r(x, y);
      BREAK;
   else H := H − {(x, y)};
   endif;
endfor;
For (x, y) ∈ A do
   If (x, y) ∈ H then c*(x, y) := f(x, y);
   else c*(x, y) := c(x, y);
   endif;
endfor;
c* is the solution of the IMF∞ problem.
```

Let's prove now the correctness of the algorithm:

Theorem 3.1 (correctness of the algorithm)**.** *The vector c^* found by the algorithm is the solution of IMF∞ and $\|c - c^*\|_\infty = d$.*

Proof. If t is accessible from s, *i.e.*, there is a directed path from s to t in the network $G' = (N, B = \widetilde{A})$, then IMF$\infty$ does not have solution (see Th. 3.1).

The algorithm continues only if IMF∞ is feasible.

The main idea of the algorithm is to change the residual capacities (by transforming them to 0) of the arcs from G_f and to leave as many as possible arcs with the greatest residual capacities unchanged. The residual capacities of the arcs from the set H will be changed to 0. The algorithm starts with $H = A_f - \widetilde{A}$, because the arcs from $\widetilde{A}$ will not be changed. The arcs from H are sorted descending by their residual capacities and so, they are prepared for elimination from H, starting with the arc with the greatest residual capacity and ending with the arc with the lowest capacity.

We shall explain now the meaning of the sets that appear in the algorithm.

The set B contains arcs with the residual capacity that will not be changed. At the beginning of the algorithm, B contains the arcs from $\widetilde{A}$. Every time accessible nodes from s are searched, the visited arcs are eliminated from the set B in order to keep the time complexity low. The search will continue from the point where it was last stopped. In this proof we shall refer to the set B as being complete, without any elimination of arcs.

At any moment, in the set W there are the nodes y which are accessible from s in the graph $G' = (N, B)$.

The set W' contains the nodes which are newly found accessible from s in the graph $G' = (N, B)$, when the current arc (x, y) enters the set B. After they are found, they are added in the set W.

From the set H the arcs of which capacities will not be changed are eliminated. At the end of the algorithm this set will help construct the solution c^* of IMF∞ (only the capacities of the arcs from H will be changed in the network G).

Now, let's see what happens with every arc (x, y) from H.

During the **for** loop, one of the following two situations can be found for every arc (x, y) of H:

Situation 1. $y \notin W$

This means that the node y was not previously found accessible from s. So, the arc (x, y) is introduced in the set B. The next time another search will be done in the network $G' = (N, B)$, this will possibly help to find other accessible nodes from s in G'. If the node x is in the set of accessible nodes from s, then y is also accessible from s in G' and it is possible that other nodes from N will be found accessible from s. So, the graph search is continued in G'. All newly found accessible nodes from W' will be added into the set W.

Situation 2. $y \in W$

It means that the node y was previously found as being accessible from s in the network $G' = (N, B)$. In this case there is no need to introduce the arc (x, y) in the set B.

The algorithm ends (if IMF∞ is feasible) when the node t enters the set B, becoming accessible from s in the graph $G' = (N, B)$. The last arc (x, y), which made the node t become accessible, will not leave the set H.

In the variable d the capacity of the first arc $(x, y) \in H$ is introduced, with the propriety that t becomes accessible from s in G'. The arc (x, y) will not be eliminated from H. So, d is the greatest value of the residual capacities of the arcs that are changed, *i.e.*, $d = max\{r(u, v) \mid (u, v) \in H\} = \max\{c(u, v) - f(u, v) \mid (u, v) \in H\} = \|c - c^*\|_\infty$. □

Theorem 3.2 (complexity of the algorithm)**.** *The algorithm for IMF∞ has a time complexity of $O(m \cdot log(n))$.*

Proof. Finding the set W with the nodes accessible from s and the elimination of the visited arcs from B can be done in linear time – $O(p)$, using a graph search algorithm applied in the graph G', where p is the number of arcs eliminated from the set B.

Every time when accessible nodes from s are searched, the arcs from B are eliminated in order to improve the time complexity of **for ... do** sequence. If they were not eliminated, then the time complexity of the algorithm would be close to $O(m^2)$, because a graph search can be done in a time complexity of $O(m)$.

Every time the current arc $(x, y) \in H$ has $x \in W$ and $y \notin W$, the initiated graph search is continued in G'.

Every arc from A_f enters B at most once and exits B at most once. It results that the total time needed to execute the whole graph search of G' is $O(m)$.

All the operations inside **for ... do** sequence (other than graph search) can be done in $O(1)$ time.

So, **for ... do** sequence is executed in linear time complexity.

The arcs from the set H can be sorted in $O(m \cdot log(m)) = O(m \cdot log(n))$ time, for instance using Mergesort. Hence, the complexity of the whole algorithm for IMF∞ depends on the method used to sort the arcs from H. □

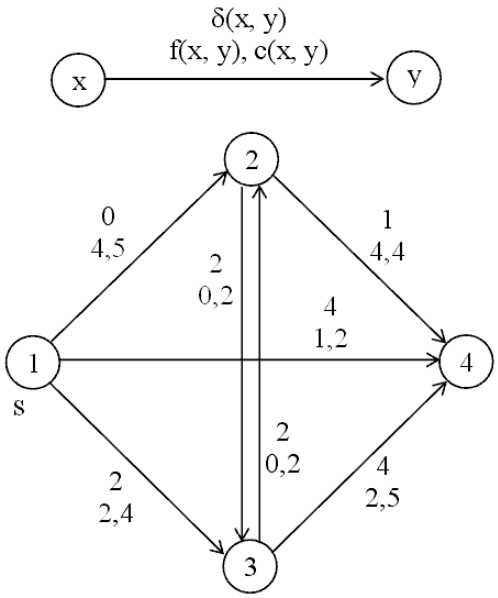

FIGURE 1. The network G and the flow f.

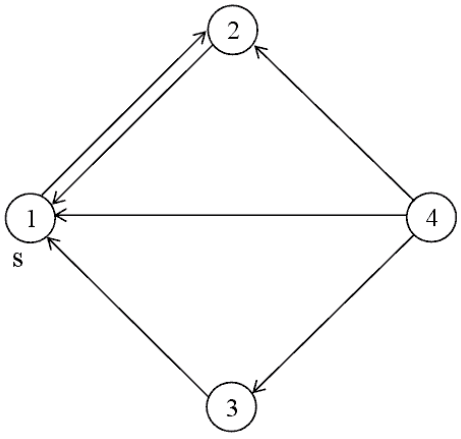

FIGURE 2. The graph $\widetilde{G} = (N, \widetilde{A})$.

4. Numerical example

We shall take a numerical example now to illustrate how the algorithm for IMF∞ works (see Fig. 1).

The graph $\widetilde{G}$ is presented in Figure 2.

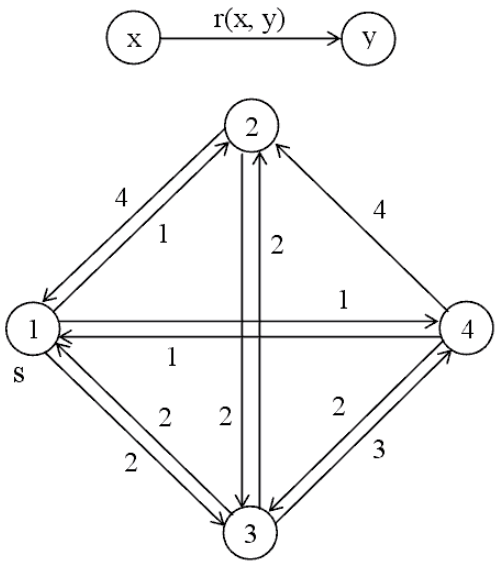

FIGURE 3. The residual network G_f.

The set of nodes accessible from $s = 1$ in the graph $G' = (N, B = \widetilde{A})$ is $W = \{1, 2\}$. The sink node $t = 4$ is not in the set W. So, IMF∞ is feasible.

The residual network G_f is presented in Figure 3.

The set H contains the arcs from A_f which are not in the set $\widetilde{A}$ (see Figs. 2 and 3), *i.e.*, $H = \{(3,4),(3,2),(2,3),(1,3),(1,4)\}$. The arcs of H are sorted descending by their residual capacities.

The iterations of the first **for ... do** sequence are:

1. $(x, y) = (3, 4), y = 4 \notin W, x = 3 \notin W$, the arc $(3, 4)$ is introduced in the set B and it is taken out of the set H.
2. $(x, y) = (3, 2), y = 2 \in W$, the arc $(3, 2)$ is taken out the set H.
3. $(x, y) = (2, 3), y = 3 \notin W, x = 2 \in W$, the arc $(2, 3)$ is introduced in the set B. The set of arcs accesible from $s = 1$ found in G' is $W' = \{3, 4\}$, $W = \{1, 2, 3, 4\}$. The node $t = 4$ is in W, $d = 2$.

The arcs of the initial network G of which capacities are modified are those left in the set $H = \{(2,3),(1,3),(1,4)\}$.

The solution of IMF∞ for the network G and the flow f is shown in Figure 4.

5. The case when lower and upper bounds for the flow can be modified

We shall consider now the inverse l_∞ maximum flow problem when in the given network G there are lower bounds for the given feasible flow f and both, lower

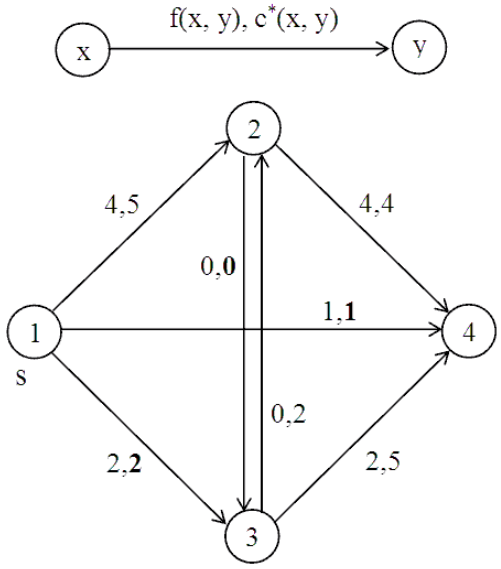

FIGURE 4. The network G^*.

and upper bounds for the flow can be modified so that f becomes a maximum flow. We shall denote this problem as IMF∞2.

In this case the capacity restrictions are:

$$l(x,y) \le f(x,y) \le c(x,y),\ \forall\ (x,y) \in A, \tag{2'}$$

where the given lower bound vector l for the flow has non-negative components.

IMF∞2 can be formulated using the following mathematical model:

$$\begin{cases} \texttt{min}\ \ \max\{\left\|l-\bar{l}\right\|_\infty, \|c-\bar{c}\|_\infty\} \\ f\ \texttt{is a maximum flow in}\ \bar{G}=\{N,A,\bar{c},\bar{l},s,t\} \\ l(x,y)-\gamma(x,y) \le \bar{l}(x,y) \le \texttt{min}\{\bar{c}(x,y), l(x,y)+\beta(x,y)\} \\ c(x,y)-\delta(x,y) \le \bar{c}(x,y) \le c(x,y)+\alpha(x,y),\ \forall\ (x,y) \in A, \end{cases} \tag{9}$$

where $\alpha(x,y), \delta(x,y), \beta(x,y)$ and $\gamma(x,y)$ are given non-negative numbers. Moreover, we have $\gamma(x,y) \le l(x,y)$ and $\delta(x,y) \le c(x,y)$, for each arc $(x,y) \in A$.

As we have seen for IMF∞, the conditions $\bar{c}(x,y) \le c(x,y)+\alpha(x,y)$, for each arc $(x,y) \in A$ have no effect. Similarly, the conditions $l(x,y)-\gamma(x,y) \le \bar{l}(x,y)$, for each arc $(x,y) \in A$ have no effect. Instead of (9), the following mathematical model is considered:

$$\begin{cases} \texttt{min}\ \ \max\{\left\|l-\bar{l}\right\|_\infty, \|c-\bar{c}\|_\infty\} \\ f\ \texttt{is a maximum flow in}\ \bar{G}=\{N,A,\bar{c},\bar{l},s,t\} \\ \bar{l}(x,y) \le \texttt{min}\{\bar{c}(x,y), l(x,y)+\beta(x,y)\} \\ c(x,y)-\delta(x,y) \le \bar{c}(x,y),\ \forall\ (x,y) \in A, \end{cases} \tag{9'}$$

We shall consider in the network G for each nodes x and y from N ($x \neq y$) that there is at most one arc that connects these two nodes, *i.e.*, there are no parallel arcs and:

$$\forall x, y \in N, x \neq y, (x, y) \in A => (y, x) \notin A. \tag{10}$$

A network with parallel arcs can be transformed into a network with no parallel arcs as follows: for each parallel arc (x, y) a new node z is introduced in the set N, the arcs (x, z) and (z, y) are introduced in the set A with $c(x, z) = c(z, y) = c(x, y)$, $l(x, z) = l(z, y) = l(x, y)$, $f(x, z) = f(z, y) = f(x, y)$ and after that the arc (x, y) is eliminated from A.

Similarly, if a network does not have the propriety (10), then it can be transformed into a network with this propriety. For each pair of nodes x and y ($x \neq y$) so that $(x, y), (y, x) \in A$, a new node z is introduced in the set N, the arcs (y, z) and (z, x) are introduced in the set A with $c(y, z) = c(z, x) = c(y, x)$, $l(y, z) = l(z, x) = l(y, x)$, $f(y, z) = f(z, x) = f(y, x)$ and the arc (y, x) is eliminated from A.

The capacities of the residual network $G_f = (N, A_f, r, s, t)$ attached to the network G with the propriety (10) and for the flow f are:

$$r(x, y) = \begin{cases} c(x, y) - f(x, y), (x, y) \in A \\ f(y, x) - l(y, x), (y, x) \in A \\ 0, \ \texttt{otherwise}. \end{cases} , \forall (x, y) \in N \times N \tag{4'}$$

Similarly to IMF∞, if $c(x, y) > f(x, y) + \delta(x, y)$ on an arc (x, y) of A, then, when solving IMF∞2, there is no need to change the upper bound on (x, y).

If the lower bound for the flow is changed on an arc (x, y), then the lower bound will be increased with the amount of $f(x, y) - l(x, y)$ on the arc (x, y). If not so, then the flow f is not stopped from being increased on a path from s to t that contains the inverse directed arc (x, y) and the modification of the lower bound is useless. This means that if $f(x, y) > l(x, y) + \beta(x, y)$ on an arc (x, y), then, when solving IMF∞2, there is no need to change the lower bound of (x, y).

Similarly to IMF∞, we consider the same set $\widetilde{A}_1$ that contains the arcs from A on which it is no use to change the upper bounds (see (5)).

The set of arcs of the graph $\widetilde{G}$ used to verify the fesibility of IMF∞2 is:

$$\widetilde{A} = \widetilde{A}_1 \cup \{(x, y) \in N \times N \,|(y, x) \in A \ \texttt{and} \ f(y, x) > l(y, x) + \beta(y, x)\}. \tag{8'}$$

Now it is clear that the algorithm for IMF∞ can be easily adapted for IMF∞2. It does not have to suffer major modifications. The algorithm has the same steps, only the sets $\widetilde{A}$ and A_f are different. These sets are constructed using the formulas (4′) and (8′), instead of (4) and (8). At the end of the algorithm, the set H contains the arcs on which the upper bounds are changed and the inverse directed arcs on which the lower bounds for the flow are changed. The last **for ... do** sequence will be rewritten as follows:

```
For (x, y) ∈ A do
   If (x, y) ∈ H then c*(x, y) := f(x, y);
```

```
    else c*(x, y) := c(x, y);
    If (y, x) ∈ H then l*(x, y) := f(x, y);
    else l*(x, y) := l(x, y);
endfor;
(l*, c*) is the solution of the IMF∞2 problem.
```

The proof of the correctness of the algorithm for IMF∞2 is similar to the proof of IMF∞. The time complexity is also the same.

We shall study now the inverse minimum flow problem (denoted ImF∞), where both, lower and upper bounds for the flow can be changed in order to make the given feasible flow become a minimum flow.

The minimum flow problem is:

$$\begin{cases} \texttt{min } v(f) \\ f \texttt{ is a feasible flow in } G. \end{cases} \tag{11}$$

We shall consider f a feasible flow in G even if the value of the flow *v(f)* is negative.

Similarly to IMF∞2, ImF∞ can be formulated as follows:

$$\begin{cases} \texttt{min } max\{\|l - \bar{l}\|_\infty, \|c - \bar{c}\|_\infty\} \\ f \texttt{ is a minimum flow in } \bar{G} = \{N, A, \bar{c}, \bar{l}, s, t\} \\ \bar{l}(x, y) \leq \texttt{min}\{\bar{c}(x, y), l(x, y) + \beta(x, y)\} \\ c(x, y) - \delta(x, y) \leq \bar{c}(x, y),\ \forall\ (x, y) \in A. \end{cases} \tag{12}$$

The minimum flow from s to t in the network G is obtained by applying a maximum flow algorithm from the sink node t to the source node s in the network G [4].

The algorithm for the inverse minimum flow problem considering l_∞ norm is adapted from the algorithm for the inverse maximum flow considering l_∞ norm, where s changes its role with t [5]. Let's prove now that ImF∞ can be solved using the algorithm for IMF∞2 applied from t to s.

Let (l^*, c^*) be an optimal solution for IMF∞2 applied from t to s. It results that f is a maximum flow from t to s (see $(9')$). This means that f is a minimum flow from s to t and because $\max\{\|l - l^*\|_\infty, \|c - c^*\|_\infty\}$ is minimum, c^* satisfies the same constrains as for IMF∞2 (see $(9')$ and (12)), it follows that (l^*, c^*) is an optimal solution for ImF∞.

6. Conclusion

An $O(m \cdot \log(n))$ time algorithm for IMF∞ has been presented. It is much faster than the algorithms for the IMF problem, because these algorithms need to find a minimum cut in an auxiliary network [12]. So, where the circumstances permit, the algorithm IMF∞ can be preferred to the algorithms for the IMF problem. Moreover, the case when both, lower and upper bounds for the flow can be modified is studied and the algorithm is adapted for this case. The inverse minimum flow problem (ImF∞) considering l_∞ norm can be solved applying the

same algorithm as for IMF∞2 from the sink node t to the source s. The inverse maximum flow and the inverse minimum flow problems can be tested in linear time and space not depending on the norm if they have solution (using a search algorithm applied to the graph $\widetilde{G}$).

References

[1] R.K. Ahuja, T.L. Magnanti and J.B. Orlin, *Network Flows. Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, NJ (1993).

[2] R.K. Ahuja and J.B. Orlin, *Combinatorial Algorithms for Inverse Network Flow Problems*, Networks (2002).

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

[4] E. Ciurea and L. Ciupala, Sequential and Parallel Algorithms for Minimum Flows, *J. Appl. Math. Comput.* **15** (2004) 53–75.

[5] E. Ciurea and A. Deaconu, Inverse Minimum Flow Problem, *J. Appl. Math. Comp., Korea* **23** (2007) 193–203.

[6] A. Deaconu, The Inverse Maximum Flow Problem with Lower and Upper Bounds for the Flow, to appear in *YUJOR* **18** (2008).

[7] A. Deaconu, A Cardinality Inverse Maximum Flow Problem, *Scientific Annals of Computer Science* **XVI** (2006) 51–62.

[8] C. Heuberger, Inverse Combinatorial Optimization: A Survey on Problems, Methods, and Results, *J. Combin. Optim.* **8** (2004) 329–361.

[9] L. Liu and J. Zhang, Inverse Maximum Flow Problems under Weighted Hamming Distance, *J. Combin. Optim.* **12** (2006) 395–408.

[10] P.T. Sokkalingam, R.K. Ahuja and J.B. Orlin, Solving Inverse Spanning Tree Problems through Network Flow Techniques, *Oper. Res.* **47** (1999) 291–298.

[11] C. Yang and X. Chen, An Inverse Maximum Capacity Path Problem with Lower Bound Constraints, *Acta Math. Sci., Ser. B* **22** (2002) 207–212.

[12] C. Yang, J. Zhang and Z. Ma, Inverse Maximum Flow and Minimum Cut Problems, *Optimization* **40** (1997) 147–170.

[13] J. Zhang and C. Cai, Inverse Problems of Minimum Cuts, *ZOR-Math. Methods Oper. Res.* **47** (1998) 51–58.