THE INVERSE MAXIMUM FLOW PROBLEM UNDER WEIGHTED

\( l_k \) NORM

Adrian DEACONU

Abstract

The problem consists in modifying the lower and the upper bounds of a given feasible flow \( f \) in a network \( G \) so that the given flow becomes a maximum flow in \( G \) and the distance between the initial vector of bounds and the modified one measured using weighted \( L_k \) norm \( (k \in \mathbb{N}) \) is minimum. We denote this problem by \( \text{IMFWL}_k \).

\( \text{IMFWL}_k \) is a generalization of the inverse maximum flow problem under \( L_k \) norm (denoted \( \text{IMFL}_k \), where the per unit cost of modification is equal to 1 on all arcs), which was previously studied and solved in polynomial time. In this paper, the algorithm for \( \text{IMFL}_k \) is adapted to solve \( \text{IMFWL}_k \).


Key words: inverse combinatorial optimization, maximum flow, strongly polynomial time complexity.

1 Introduction

The inverse combinatorial optimization problems are relatively new and they have been studied intensively in the last years. 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 \( l_1 \), \( l_\infty \) 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 [5]. Almost every inverse problem was studied considering \( l_1 \) and \( l_\infty \) norms, resulting in different problems with completely different solution methods. Strongly polynomial time algorithms to solve the inverse maximum flow problem under \( l_1 \) norm (denoted \( \text{IMF} \)) were presented in [9] and, then in [3]. \( \text{IMF} \) is reduced to a minimum cut problem in an auxiliary network. Four inverse maximum flow problems are also studied by Liu and Zhang [6] under the sum-type and bottleneck-type weighted Hamming distances. Strongly polynomial algorithms for these problems are proposed.

\( ^1 \)Faculty of Mathematics and Informatics, Transilvania University of Braşov, Romania, e-mail: a.deaconu@unitbv.ro or am_deaconu@yahoo.com
In this paper, the inverse maximum flow considering weighted $l_k$ norm (denoted IMFWL$_k$) is studied. The inverse maximum flow problem under $l_k$ norm (denoted IMFL$_k$) was studied and solved in [4]. In this paper we show how the algorithm for IMFL$_k$ can be adapted to solve the more general problem: IMFWL$_k$.

2 The IMFWL$_k$ 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 a source or/and more than a sink node, it can be transformed into a 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 flow $f$ is:

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

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

The capacity restrictions are:

$$0 \leq f(x,y) \leq c(x,y), \forall (x,y) \in A.$$

The maximum flow problem is:

$$\max v(f) \text{ subject to } \begin{cases} f \text{ is a feasible flow in } G \end{cases}.$$

The residual network attached to network $G$ for 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:

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

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 under weighted $l_\infty$ norm is to change the capacity vector $c$ so that the given feasible flow $f$ becomes a maximum flow in $G$ and the maximum cost of change of the capacities on arc is minimum.

IMFWL$_k$ can be formulated using the following mathematical model:
Moreover, this test of feasibility can be applied to any inverse maximum flow problem

\[ f \text{ is a maximum flow in } \tilde{G} = \{N, A, \tilde{l}, s, t\} \]

\[ l(x, y) - \gamma(x, y) \leq \tilde{l}(x, y) \leq \min \{\tilde{c}(x, y), l(x, y) + \beta(x, y)\} \]

\[ c(x, y) - \delta(x, y) \leq \tilde{c}(x, y) \leq c(x, y) + \alpha(x, y), \forall (x, y) \in A \]

(5)

where \( w_l(x, y) > 0 \) and \( w_c(x, y) > 0, \forall (x, y) \in A \).

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

In order to make flow \( f \) become a maximum flow in network \( G \), the upper bounds of some arcs from \( A \) must be decreased. So, conditions \( \tilde{c}(x, y) \leq c(x, y) + \alpha(x, y) \), for each arc \( (x, y) \in A \) have no effect. Similarly, the lower bounds of some arcs from \( A \) must be increased and, so, conditions \( l(x, y) - \gamma(x, y) \leq \tilde{l}(x, y) \), for each arc \( (x, y) \in A \) have no effect. Instead of (5), the following mathematical model is considered:

\[
\begin{align*}
& \min \sum_{(x,y) \in A} w_l(x,y) |l(x, y) - \tilde{l}(x, y)|^k + w_c(x,y) |c(x,y) - \tilde{c}(x,y)|^k \\
& f \text{ is a maximum flow in } \tilde{G} = \{N, A, \tilde{l}, s, t\} \\
& \tilde{l}(x, y) \leq \min \{\tilde{c}(x, y), l(x, y) + \beta(x, y)\} \\
& c(x, y) - \delta(x, y) \leq \tilde{c}(x, y), \forall (x, y) \in A
\end{align*}
\]

(6)

A graph denoted \( \tilde{G} = (N, \tilde{A}) \) can be constructed to verify the feasibility of IMFWL\(_k\) (see [4]), where:

\[
\tilde{A} = \{(x, y) \in A | c(x,y) > f(x,y) + \delta(x, y)\} \cup (7)
\]

\[
\cup \{(x, y) \in N \times N | (y, x) \in A \text{ and } f(y,x) > l(y,x) + \beta(y, x)\}.
\]

We have the following theorem:

**Theorem 1.** In network \( G \), IMFWL\(_k\) has optimal solution for the given flow \( f \), if and only if there is no directed path in the graph \( G \) from the node \( s \) to the node \( t \).

**Proof.** see [4].

The verification of IMFWL\(_k\) being feasible can be done in \( O(p) \) time complexity, using a graph search algorithm in \( \tilde{G} \), where \( p \) is the number of arcs in the set \( \tilde{A} \) with \( p \leq m \). Moreover, this test of feasibility can be applied to any inverse maximum flow problem (under any norm).
3 Algorithm for IMFWL\textsubscript{k}

**Definition 1.** The set of arcs \([X, \bar{X}]\) is called an \(s-t\) cut in network \(G\), where: \(X \subset N, \ X \subset N, \ s \in X, \ t \in \bar{X}\) and
\[
(X, \bar{X}) = \{(x, y) \in A | x \in X \text{ and } y \in \bar{X}\}
\]
(8)
\[
(\bar{X}, X) = \{(x, y) \in A | x \in \bar{X} \text{ and } y \in X\}.
\]
(9)

\((X, \bar{X})\) is called the set of direct arcs of the \(s-t\) cut and \((\bar{X}, X)\) is called the set of inverse arcs of the \(s-t\) cut.

**Definition 2.** The capacity of the \(s-t\) cut \([X, \bar{X}]\) is \(c[X, \bar{X}] = c(X, \bar{X}) - l(\bar{X}, X) = \sum_{(x,y) \in (X, \bar{X})} c(x,y) - \sum_{(x,y) \in (\bar{X}, X)} l(x,y)\).

**Definition 3.** An \(s-t\) cut \([X, \bar{X}]\) is a minimum \(s-t\) cut if its capacity is minimum, i.e., \(c[X, \bar{X}] = \min\{c[X', \bar{X}'][X', \bar{X}'] \text{ is an } s-t \text{ cut}\}\).

Starting from the residual network \(G_f\) we construct the network \(G^k_f = (N, A_f, r^k_w, s, t)\), where for any \((x,y) \in A_f\) we have:
\[
r^k_w(x,y) = \begin{cases} +\infty, & (x,y) \in \tilde{A} \\
 w_l(x,y)(l(x,y) - \bar{l}(x,y))^k + w_c(x,y)(c(y,x) - \bar{c}(y,x))^k, & \text{otherwise}
\end{cases}
\]
(10)

In network \(G^k_f\) we compute a minimum \(s-t\) cut \([X, \bar{X}]\). We denote by \(B\) the set of direct arcs of this minimum cut, i.e., \(B = (X, \bar{X})\).

The solution of IMFWL\textsubscript{k} is vector \((l^*, c^*)\) (see [4]), where:
\[
l^*(x,y) = \begin{cases} f(x,y), & \text{if } (y,x) \in B \\
 l(x,y), & \text{otherwise}
\end{cases}, (x,y) \in A.
\]
(11)
\[
c^*(x,y) = \begin{cases} f(x,y), & \text{if } (x,y) \in B \\
 c(x,y), & \text{otherwise}
\end{cases}, (x,y) \in A.
\]
(12)

The algorithm for solving IMFWL\textsubscript{k} is as follows:

**Step 1:**
Construct the graph \(\tilde{G} = (N, \tilde{A})\) (see (7));
If there is a directed path in \(\tilde{G}\) from \(s\) to \(t\)
then IMFWL\textsubscript{k} does not have solution; STOP.
else goto step 2.

**Step 2:**
Construct the residual network \(G_f = (N, A_f, r, s, t)\) (see (4));
Construct network $G_f^k = (N, A_f, r^k_w, s, t)$ (see (10));
Find a minimum $s-t$ cut $[X, \bar{X}]$ in $G_f^k$;
B:=($X$, $\bar{X}$);
Construct vector $l^*$ using (11);
Construct vector $c^*$ using (12);

Vector $(l^*, c^*)$ is the optimum solution of IMFWL$_k$.

The upper and lower bounds $l$ and $c$ of the initial network $G$ are substituted by $l^*$ and, respectively, $c^*$ in order to obtain network $G^*$, in which $f$ is a maximum flow.

**Theorem 2.** The strongly polynomial implementation of the algorithm for solving IMFWL$_k$ has a time complexity of $O(n \cdot m \cdot \log(n^2/m))$, where $n = |N|$ and $m = |A|$.

*Proof.* The proof is similar to the proof of time complexity of the algorithm for IMFL$_k$ (see [4]).

In some cases, when the capacities are small, it is better to apply a weakly polynomial algorithm to find the minimum $s-t$ cut. Weakly polynomial (complexity of which depends on $\log C$) and non-polynomial (complexity of which depends on $C$) algorithms for minimum cut can not be applied directly to solve IMFWL$_k$ because the capacity of some arcs in $G_f^k$ can have infinite values and the time complexity of these algorithms depends on the maximum value of the capacities of the arcs.

**Theorem 3.** The time complexity of the weakly polynomial implementation of the algorithm for solving IMFWL$_k$ is $O(\min\{n^{2/3}, m^{1/2}\} \cdot m \cdot \log(n^2/m) \cdot \log(\max\{n, R\}))$, where:

$$R = \max\{r^k_w(x, y) | (x, y) \in A_f\}. \quad (13)$$

*Proof.* The proof is similar to the proof of time complexity of the algorithm for IMFL$_k$ (see [4]).

4 Some particular cases

There are many cases when the initial network $G$ has only upper bounds (capacities) for the flow. In this case the lower bounds are all equal to 0 and they can not be modified. This can be treated as a particular case of IMFWL$_k$. Indeed, in this case we can consider on each arc $(x, y) \in A$ that $\beta(x, y) = 0$.

Similarly, if only the lower bounds for the flow can be modified, this problem can be also treated as a particular case of IMFWL$_k$. In this case we can consider on each arc $(x, y) \in A$ that $\delta(x, y) = 0$. 
5 Conclusion

In this paper we extended the result from [4] under weighted $L_k$ norm, where $k$ is a positive number. A very fast method (in linear time) to verify apriori the feasibility of IMFWL$_k$ can be applied. There is no need to apply the algorithm for minimum cut in network $G_j^k$ if IMFWL$_k$ is not feasible. We presented an adaptation of the algorithm for IMFL$_k$ to solve IMFWL$_k$. We showed how this algorithm can be implemented in strongly or weakly polynomial time. For each case, the time complexity has been presented. If the initial network $G$ does not have lower bounds, this can be treated as a particular case of IMFWL$_k$.

References


