Journal of Mathematical Sciences: Advances and Applications
Volume 43, 2017, Pages 1-18
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/jmsaa_7100121736

ALGORITHM OF THE INVERSE MINIMUM FLOW PROBLEM WITH TIME-WINDOWS

NASSER A. EL-SHERBENY

Department of Mathematics
Faculty of Science
Al-Azhar University
Nasr City (11884)
Cairo
Egypt
e-mail: nasserelsherbeny@yahoo.com

Department of Mathematics
Faculty of Applied Medical Science
Taif University
Turabah
Kingdom of Saudi Arabia

Abstract

In this paper, we present the inverse minimum flow problem with time-windows, where lower and upper bounds for the flow must be changed as little as possible so that a given feasible flow becomes a minimum flow that satisfies a time-windows constraint for each vertex in the network. A linear time and space method to decide if the problem has solution is presented. The inverse minimum flow problem with time windows has the inverse combinatorial optimization and an NP-hard problem. We propose in this work a new algorithm for solving the 'inverse minimum flow problem with time-windows' (IMFPTW).

2010 Mathematics Subject Classification: 05C35, 68R10, 90C27, 90C35.
Keywords and phrases: inverse combinatorial optimization, optimization network, minimum flow, graph search, time-windows.
Received November 9, 2016

© 2017 Scientific Advances Publishers

1. Introduction

In recent years, many papers were published in the field of the inverse combinatorial optimization [2], [3], [4], [5], and [12]. The field of network optimization flows has a rich and long history. An inverse combinatorial optimization problem consists of 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. The inverse maximum flow problem, which is related to the inverse minimum flow problem, has been studied by Yang et al. [16]. Strongly polynomial algorithms to solve the inverse maximum flow were presented.

In [16], only the upper bounds for the flow are changed as little as possible in order to make a given feasible flow become a maximum flow. That is why in many networks the inverse maximum flow problem does not have solution. In this work, we study the inverse minimum flow problem with time-windows (IMFPTW), where both, lower and upper bounds for the flow must be changed in order to make a given flow become a minimum flow satisfy a time-windows constraint. If there are not many restrictions in modifying the bounds for the flow, then the inverse minimum flow problem with time-windows always has solution. The cases when only the lower bounds and only the upper bounds can be changed in order to transform a feasible flow into a minimum flow and other cases are also studied. As it will be seen, these can be treated as particular cases of the inverse minimum flow problem with time-windows.

Let $G = (V, A, u, l, s, \tau)$ be a network, where V is the set of vertices, A is the set of directed arcs, u is the upper bound for the flow vector, l is the lower bound for the flow vector, s is the source vertex, and τ is the sink vertex. Each arc has a nonnegative transit time t_{ij}; $i \neq j$; $i, j = 1, 2, \ldots, n$. For each vertex $i \in V$, a time-windows $[a_i, b_i]$ within which the vertex may be served and $t_i \in [a_i, b_i]$, $t_i \in T$ is a nonnegative

service and leaving time of the vertex. A source vertex s, with time windows $[a_s, b_s]$, a sink vertex τ with time-windows $[a_\tau, b_\tau]$ and t_s is a departure time of the source vertex (see [7], [8], [9], [10], [11], and [15]).

The reminder of this work is organized as follows. In Section 2, we give the mathematical formulation model of the IMFPTW, basic concepts and some definitions. In Section 3, we present the algorithm for the IMFPTW. In Section 4, we give a particular case problem. In Section 5, we present an application instance. Finally, the conclusion is given in Section 6.

2. Mathematical Model of the IMFPTW

Consider $G = (V, A, u, l, s, \tau)$ be an $s - \tau$ network, where V is the set of vertices, A is the set of directed arcs, u is the upper bound, and l is the lower bound for the flow vector, where $l \leq u$. A single source vertex s, a single sink vertex τ with time windows $[a_s, b_s]$, $[a_\tau, b_\tau]$, respectively. Each arc has a nonnegative transit time t_{ij}; $i \neq j$; $i, j = 1, 2, \ldots, n$. For each vertex $i \in V$, a time-windows $[a_i, b_i]$ within which the vertex may be served and $t_i \in [a_i, b_i]$, $t_i \in T$ is a nonnegative service and leaving time of the vertex.

Let f be a given feasible flow defined by a function $f : A \to \Re^+$ in the $s - \tau$ network G, a feasible flow has to satisfy the flow balance, the time-windows and the capacity conditions:

- The balance condition for the flow f is;

$$\sum_{(i,j)\in A,\, j\in V} f_{ij} - \sum_{(j,i)\in A,\, j\in V} f_{ji} = \left\{\begin{matrix} v(f), & i = s \\ -v(f), & i = \tau \\ 0, & i \in N - \{s, \tau\} \end{matrix}\right\}, \tag{1}$$

where $v(f)$ is the value of the flow f from a source vertex s to a sink vertex τ. We consider f to be a feasible flow even if the value of the flow $v(f)$ is negative.

• The time-windows condition are:

$$t_i + t_{ij} \leq t_j, \quad t_i \in [a_i, b_i], \quad t_j \in [a_j, b_j], \quad t_i, t_j, t_{ij} \in T \in \Re^+, \quad i \neq j, \forall i, j \in V,$$

$$i, j = 1, 2, \ldots, n. \tag{2}$$

• The capacity condition are:

$$l_{ij} \leq f_{ij} \leq u_{ij}, \forall (i, j) \in A. \tag{3}$$

The minimum flow problem with time-windows is given by:

$$\min v(f)$$

$$f \text{ is a feasible flow in the network } G, \tag{4}$$

which satisfies the conditions (1), (2), and (3).

A theorem for the minimum flow problem with time-windows can be formulated in a similar way as for the theorem of the maximum flow, see [1].

Theorem 1. *A feasible flow f is a minimum flow time-windows in the network* $G = (V, A, u, l, s, \tau)$ *if and only if there is no path* $P = (s = x_1, x_2, \ldots, x_n = \tau)$ *from the source vertex s to the sink vertex* τ *so that for each arc* $(x_i, x_{i+1}) \in P : l(x_i, x_{i+1}) \prec f(x_i, x_{i+1})$ *or* $f(x_{i+1}, x_i) \prec u(x_{i+1}, x_i)$ *with* x_i, x_{i+1} *has a time windows* $[a_{x_i}, b_{x_i}], [a_{x_{i+1}}, b_{x_{i+1}}]$, *respectively.*

Note. For each arc $(x_i, x_{i+1}) \in P$ in a network $G = (V, A, u, l, s, \tau)$ a time-windows constraint can be represented by see Figure 1:

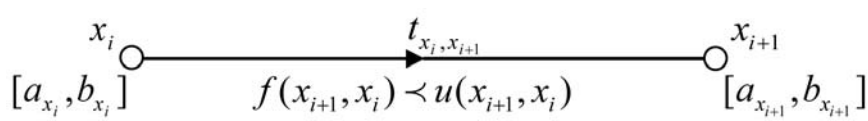

Figure 1. A representation time-windows.

Definition 1. The inverse minimum flow problem with time-windows (IMFPTW) is to change the lower bound vector l and the upper bound vector u as little as possible flow f become a minimum flow time-windows in a network.

• A mathematical model of the IMFPTW can be formulated by the following:

$$\min\|l - \bar{l}\| + \|u - \bar{u}\|$$

$$f \text{ is a minimum flow with time-windows in } \overline{G} = (V, A, \bar{u}, \bar{l}, s, \tau)$$

$$l_{ij} - \gamma_{ij} \leq \bar{l}_{ij} \leq \min\{\bar{u}_{ij}, l_{ij} + \beta_{ij}\}$$

$$u_{ij} - \delta_{ij} \leq \bar{u}_{ij} \leq u_{ij} + \alpha_{ij}, \forall (i, j) \in A$$

$$t_i + t_{ij} \leq t_j, \quad t_i \in [a_i, b_i], t_j \in [a_j, b_j], \quad t_i, t_j, t_{ij} \in T \in \Re^+, i \neq j,$$

$$\forall i, j \in V, \quad i, j = 1, 2, \ldots, n, \tag{5}$$

where $\alpha_{ij}, \beta_{ij}, \delta_{ij}, \gamma_{ij}, t_i, t_j$, and t_{ij} are nonnegative numbers. Moreover, we have $\gamma_{ij} \leq l_{ij}$ and $\delta_{ij} \leq u_{ij}$, for each arc $(i, j) \in A$.

• In the IMFPTW, we observe that if the lower bound is changed on an arc $(i, j) \in A$, then it will be increased with the amount of $f_{ij} - l_{ij}$. If not so, then the flow f is not stopped from being decreased on a path from s to τ that contains the direct arc (i, j) and the modification of the lower bound is useless. This means that if $l_{ij} + \beta_{ij} \prec f_{ij}$ on an arc (i, j) then, when solving the IMFPTW the lower bound will not be changed on the arc (i, j).

• Similarly, if the upper bound is changed on an arc (i, j), then it will be decreased with the amount of $u_{ij} - f_{ij}$ in order to stop the flow from being decreased on a path from s to τ that contains the arc (i, j) in

inverse direction. So, if $u_{ij} - \delta_{ij} \succ f_{ij}$ on the arc (i, j), then the value u_{ij} of the upper bound will not be changed on the arc (i, j). It is easy to see that the IMFPTW has solution if and only if there is no path from s to τ that contains direct arcs (i, j) with $l_{ij} + \beta_{ij} \prec f_{ij}$ and (or) inverse directed arcs (i, j) with $u_{ij} - \delta_{ij} \succ f_{ij}$, because on such a path (if exist) the flow f cannot be stopped from being decreased by modifying the bounds of the arcs.

So, a graph denoted $\overline{G} = (V, \overline{A})$ can be constructed to verify if IMFPTW has solution, where

$$\overline{A} = \{(i, j) : l_{ij} + \beta_{ij} \prec f_{ij} \text{ or } u_{ji} - \delta_{ji} \succ f_{ji}\}. \tag{6}$$

Lemma 1. *In the network* $G = (V, A, u, l, s, \tau)$, *the IMFPTW has solution for the given flow f if and only if there is no directed path in the graph* $\overline{G} = (V, \overline{A})$ *from the vertex s to the vertex* τ.

The verification can be done in a complexity of $O(\overline{m})$, using a graph search algorithm in $\overline{G} = (V, \overline{A})$, where $m, \overline{m}$ are the number of arcs in the set A and the number of arcs in the set $\overline{A}$, respectively, where $\overline{m} \leq m$. Of course, if the set $\overline{A}$ is empty, then the IMFPTW has solution.

Note. The minimum flow problem from the vertex s to the vertex τ in the network $G = (V, A, u, l, s, \tau)$ is obtained by applying a maximum flow algorithm from the vertex τ to the vertex s in the network G. The minimum flow problem can be solved by using a maximum flow algorithm, see [6].

Theorem 2. *A feasible flow f in the network* $G = (V, A, u, l, s, \tau)$ *is a minimum flow from the vertex s to the vertex* τ *if and only if there is no path from the vertex* τ *to the vertex s that can increase the flow from* τ *to s.*

The solving of the IMFPTW is to change as little as possible the bounded vectors l and u for the flow so that f becomes a maximum flow time-windows from τ to s in the network G, where the value of the flow f can be even negative.

We introduce the definition of the residual network of G for the flow f, we consider now that in the network G for each arc $(i, j) \in A$ there is an arc $(j, i) \in A$. If not so, then for each arc $(i, j) \in A$ so that $(j, i) \notin A$ it can be introduced the arc $(i, j) \in A$ with $u_{ji} = l_{ji} = 0$.

Now, we consider the residual network $G_f = (V, A_f, r, s, \tau)$ attached to the network G for the flow f, where

$$r_{ij} = u_{ij} - f_{ij} + f_{ji} - l_{ji}, \forall (i, j) \in A. \tag{7}$$

The set A_f contains the arcs (i, j) for which the residual capacity is positive; $r_{ij} \succ 0$. It is obviously that in order to find the minimum flow f, the maximum flow must be found using the residual network $G_f = (V, A_f, r, \tau, s)$, where the source vertex s changes its role with the sink vertex τ.

Consider the network $G^{/} = (V, A^{/}, u^{/}, l^{/}, s, \tau)$, where $A^{/} = A_f$, $u^{/} = r$, and $l^{/} = 0$. The set of arcs $[X, \overline{X}] = (X, \overline{X}) \cup (\overline{X}, X)$ is a $\tau - s$ cut in $G^{/}$ if $X \cap \overline{X} = \phi$, $X \cup \overline{X} = V$, $\tau \in X$ and $s \in \overline{X}$, where $(X, \overline{X}) = \{(i, j) \in A^{/} : i \in X \wedge y \in \overline{X}\}$ is the set of directed arcs of the cut and $(\overline{X}, X) = \{(i, j) \in A^{/} : i \in \overline{X} \wedge j \in X\}$ is the set of the inverse arcs.

The capacity of the $\tau - s$ cut $[X, \overline{X}]$ in $G^{/}$ is

$$u^{/}[X, \overline{X}] = u^{/}(X, \overline{X}) = \sum_{(i, j) \in (X, \overline{X})} u_{ij}^{/}. \tag{8}$$

A $\tau - s$ cut is a minimum cut in $G^{/}$ if its capacity is minimal in the set of $\tau - s$ cuts in the network $G^{/}$. If the set $\widetilde{A} = \phi$, then the solving of the IMFPTW is equivalent to find the set of arcs τ so that if the arcs of B are eliminated from A_f, then there is no directed path in $G_f^B = (V, A_f - B)$ from the sink vertex τ to the source vertex s and $r(B) = \sum_{(i,j)\in B} r_{ij}$ is minimal. This means that B is the set of direct arcs of a minimal $\tau - s$ cut in the network $G^{/}$.

As we have seen, if the set $\widetilde{A} \neq \phi$, then when solving the IMFPTW, no change will be done to the lower bounds and/or to the upper bounds on the arcs of $(i, j) \in A$ with;

$$l_{ij} + \beta_{ij} \prec f_{ij} \text{ or } u_{ij} - \delta_{ij} \succ f_{ij}. \tag{9}$$

This means that for each arc $(i, j) \in A$ so that $l_{ij} + \beta_{ij} \prec f_{ij}$ the residual capacity of (j, i) can be set to ∞ and for each arc $(i, j) \in A$ such that $u_{ij} - \delta_{ij} \succ f_{ij}$ the residual capacity of (i, j) can also be modified to ∞. If the IMFPTW has solution, then by setting the bounds to ∞, we assure that these arcs will not be in the set B, which is the set of direct arcs of the minimum $\tau - s$ cut in the network $G^{/} = G_f$.

So if the set $\widetilde{A} \neq \phi$, then a minimum $\tau - s$ cut must be searched in the network $G^{//} = (V, A^{//} = A_f, u^{//}, \tau, s)$, where, for every arc $(i, j) \in A$;

$$u_{ij}^{//} = \begin{cases} \infty, & (j, i) \in \widetilde{A}, \\ r_{ij}, & \text{otherwise.} \end{cases} \tag{10}$$

The set of direct arcs of the minimum $\tau - s$ cut in $G^{//}$ is denoted B. If the upper bounds of the arcs $(i, j) \in A \cap B$ are decreased with the quantity $u_{ij} - f_{ij}$ and the lower bounds of the arcs $(i, j) \in A$ with $(j, i) \in B$ are

increased with the quantity $f_{ij} - l_{ij}$, then the flow is stopped from being decreased from s to τ and it becomes a minimum flow time-windows in the network G with the modified bounds for the flow. So, for each arc $(i, j) \in A \cap B$, the upper bound must be changed to the value f_{ij} and for each arc $(i, j) \in A$, where $(j, i) \in B$, the lower bound of (i, j) must be changed to the value f_{ij}. This means that the solution of the IMFPTW is the pair of vectors (u^*, l^*), where, for every $(i, j) \in A$;

$$u_{ij}^* = \begin{cases} f_{ij}, & (i, j) \in B, \\ u_{ij}, & \text{otherwise;} \end{cases} \tag{11}$$

$$l_{ij}^* = \begin{cases} f_{ij}, & (j, i) \in B, \\ l_{ij}, & \text{otherwise;} \end{cases} \tag{12}$$

$$t_i + t_{ij} \le t_j,\ a_i \le t_i \le b_i,\ a_j \le t_j \le b_j,\ t_i, t_j, t_{ij} \in T,\ i \ne j,\ \forall i, j \in V. \tag{13}$$

Consider $[\overline{X}, X]$ be the $s - \tau$ cut in the network G, where X and $\overline{X}$ are the sets of vertices corresponding the minimum $s - \tau$ cut in the network $G^{//}$, of which set of direct arcs is $B = (X, \overline{X})$. Then, the vectors that gives the solution of the IMFPTW, u^* and l^*, can be easier defined as follows, for every $(i, j) \in A$ and satisfy the condition (13):

$$u_{ij}^* = \begin{cases} f_{ij}, & (i, j) \in (X, \overline{X}), \\ u_{ij}, & \text{otherwise;} \end{cases} \tag{14}$$

$$l_{ij}^* = \begin{cases} f_{ij}, & (j, i) \in (\overline{X}, X), \\ l_{ij}, & \text{otherwise.} \end{cases} \tag{15}$$

Theorem 3. *The pair* (u_{ij}^*, l_{ij}^*) *is the solution for the IMFPTW* (5), *where* u_{ij}^* *and* l_{ij}^* *are defined in* (14) *and* (15), *and the total amount of change to the bound vectors is equal to the capacity of the minimum* $s - \tau$ *cut in* $G^{//} = (V, A^{//}, u^{//}, \tau, s)$, *then*

$$\left\| l_{ij}^* - l_{ij} \right\| + \left\| u_{ij} - u_{ij}^* \right\| = u^{//}(B). \tag{16}$$

Proof. Let the value of the flow $v(f)$ from s to τ in the network $G^{//}$ is

$$v(f) = \sum_{[\overline{X}, X]} f_{ij} = \sum_{(\overline{X}, X)} f_{ij} - \sum_{(X, \overline{X})} f_{ij} = \sum_{(\overline{X}, X)} l_{ij}^* - \sum_{(X, \overline{X})} u_{ij}^*. \tag{17}$$

The last equality is a consequence of (14) and (15). This means that the flow f is a minimum flow time-windows from the vertex s to the vertex τ cut in the network. $G^{//}$. So the pair (u_{ij}^*, l_{ij}^*) is a feasible solution for the problem (5). The pair (u_{ij}^*, l_{ij}^*) is an optimal solution for the problem (5) because it was obtained from the minimum $\tau - s$ cut in the network $G^{//}$ and the total amount of change to the bound vectors is equal to the capacity of the minimum $\tau - s$ cut, then

$$\begin{aligned} \left\| l_{ij}^* - l_{ij} \right\| + \left\| u_{ij} - u_{ij}^* \right\| &= \sum_{(i,j) \in A} \left| l_{ij}^* - l_{ij} \right| + \sum_{(i,j) \in A} \left| u_{ij} - u_{ij}^* \right| \\ &= \sum_{(i,j) \in (\overline{X}, X)} (f_{ij} - l_{ij}) + \sum_{(i,j) \in (X, \overline{X})} (u_{ij} - f_{ij}) = u^{//}(B). \end{aligned} \tag{18}$$

3. The Algorithm for the Inverse Minimum Flow Problem with Time-Windows

In this section, we give a new algorithm to solve the IMFPTW. As it has been seen so far, after verifying if the IMFPTW has solution, the IMFPTW can be reduced to a minimum $\tau - s$ cut problem in the network $G^{//}$.

● **The algorithm general framework for the IMFPTW can be stated by the following steps:**

(1) Construct the graph $\overline{G} = (V, \overline{A})$, with each vertex $i, j \in V$, has a time-windows $[a_i, b_i]$, $[a_j, b_j]$, respectively, and $t_i + t_{ij} \leq t_j$, $a_i \leq t_i \leq b_i$, $a_j \leq t_j \leq b_j$, $t_i, t_j, t_{ij} \in T \in \Re^+$, $i \neq j$, $i, j = 1, 2, \ldots, n$.

(2) If there is a directed path in $\overline{G}$ from the vertex s to the vertex τ, then the IMFPTW does not have solution. STOP;

Else, go to Step 3.

(3) Construct the network $G^{//} = (V, A^{//}, u^{//}, \tau, s)$;

Find the minimum $\tau - s$ cut in the network $G^{//}$;

Compute B = the set of direct arcs of the minimum $\tau - s$ cut in $G^{//}$;

Compute the pair (u_{ij}^*, l_{ij}^*), $i \neq j$, $i, j = 1, 2, \ldots, n$ has the solution of the IMFPTW.

The flow f is a minimum flow in the network $G^* = (V, A, u_{ij}^*, l_{ij}^*, s, \tau)$ constructed by the algorithm.

■ **We describe the algorithm above as the following:**

● The algorithm above finds the solution of the IMFPTW, because, in the Step 1 of the algorithm, the IMFPTW is tested if it has solution using the graph search in $\overline{G}$, see Lemma 1.

● If the IMFPTW has solution, then in the Step 3, a solution is constructed, see Theorem 3.

● The time complexity of the algorithm of the IMFPTW is given by the complexity of the method used to find the minimum cut in the network $G^{//}$.

(i) A strongly polynomial algorithm for minimum cut can be applied. For instance, the algorithm of Goldberg and Tarjan ([14]) can be considered. It has the time complexity of $O(nm\log(n^2/m))$, where $n = |V|$, $m = |A_f|$.

(ii) Weakly polynomial algorithms for minimum cut cannot be applied directly, because there can be arcs with infinite capacities in the set A_f. It is not necessarily to set the capacities of these arcs to ∞. They can be set to a value big enough. It is sufficient to set the capacity of these arcs to the value of the maximum flow from τ to s in the network $G^{//}$. If the IMFPTW has solution, then the value of the minimum flow in $G^{//}$ is not greater than $m\vartheta$, where

$$\vartheta = \max\{u_{ij} - f_{ij} + f_{ji} - l_{ji} : i, j \in V\}. \tag{19}$$

The weakly polynomial algorithm for minimum cut due to Goldberg and Rao ([13]) is applied in the network $G^{//}$, then the time complexity of the algorithm for the IMFPTW is $O(\min\{n^{2/3}, m^{1/2}\}m\log(n/m)\log\alpha))$, where $\alpha = \max\{r_{ij} : (i, j) \in A_f\} = m\vartheta$. So, the time complexity of the weakly polynomial algorithm for IMFPTW is $O(\min\{n^{2/3}, m^{1/2}\}$ $m\log(n^2/m)\log(\max\{n, \vartheta\})$. Of course, if the set $\overline{A} = \varphi$, i.e., is empty, then the time complexity of the algorithm is even less, it is,

$$O(\min\{n^{2/3}, m^{1/2}\}m\log(n^2/m)\log\vartheta). \tag{20}$$

4. A Particular Case Problem

• We consider that, if we do not want to change the lower bound of an arc $(i, j) \in A$, where $i, j \in V$ with a time-windows $[a_i, b_i], [a_j, b_j]$, respectively, then it is sufficient to set the value of $\beta_{ij} = 0$ for this arc. If we do not want to change the upper bound for the arc $(i, j) \in A$, then it is sufficient to set the value $\delta_{ij} = 0$.

If there is no restriction to change the lower bound of an arc $(i, j) \in A$, then the value of β_{ij} for this arc must be considered by $\beta_{ij} \geq f_{ij} - l_{ij}$. Similarly, if there is no restriction to change the upper bound of an arc $(i, j) \in A$, then the value of δ_{ij} for the arc (i, j) considered by $\delta_{ij} \geq u_{ij} - f_{ij}$.

• A very important of the application case of the IMFPTW is the situation when there is no restriction to change the bounds of the arcs of the network G. This means that the value of β_{ij} and δ_{ij} for all arcs $(i, j) \in A$, $i \neq j$; $i, j = 1, 2, \ldots, n$ can be set to the value of ∞.

In some situations, it is possible that only the lower bounds to be allowed to be changed in order to transform the given flow f into a minimum flow, then in this case, the mathematical model can be formulated by the following;

$$\min \|l - \bar{l}\|$$

$$f \text{ is a minimum flow with time-windows in } \overline{G} = (V, A, u, \bar{l}, s, \tau)$$

$$\bar{l}_{ij} \leq \min\{u_{ij}, l_{ij} + \beta_{ij}\}, \forall (i, j) \in A$$

$$t_i + t_{ij} \leq t_j, t_i \in [a_i, b_i], t_j \in [a_j, b_j], t_i, t_j, t_{ij} \in T, i \neq j, \forall i, j \in V. \tag{21}$$

Then, we can see that the IMFPTW considering for a modification only the lower bounds for the flow denoted by LBIMFPTW can be treated as a particular case of the IMFPTW, if in (5) the values δ_{ij} are considered to equal to 0, for every arc $(i, j) \in A$. This means that the upper bounds cannot be change.

Also, another particular problem of (5), the upper bound for the flow denoted by UBIMFPTW, can be formulated by; try to transform the given feasible flow f into a minimum flow by changing only the upper bounds.

In this case, in (5), the values β_{ij} can be considered to be equal 0, for every arc $(i, j) \in A$ and so, the lower bounds are not allowed to be modified.

Of course, for the last two particular cases problems, the set of the problems without solution is larger.

5. Application Instance

We given an application instance to illustrate how the above algorithm can be applying, for each arc $(i, j) \in A$, see Figure 2.

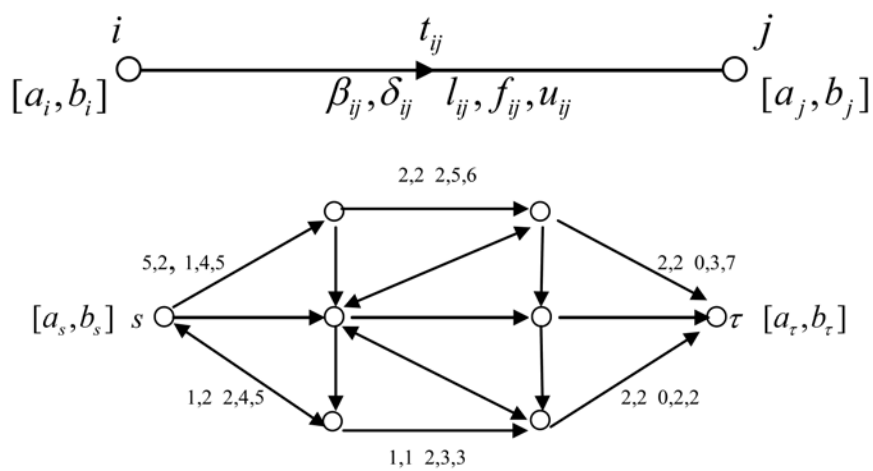

Figure 2. A flow f on a network $G = (V, A, u, l, s, \tau)$ with time-windows.

We note that $\overline{A} = \{(s, 1), (1, 2), (4, 2), (3, 6), (6, 2), (5, \tau), (\tau, 6), (6, \tau)\}$. The presented graph $\overline{G}$ is given by Figure 3. Also, there is no directed path from a source vertex s to a sink vertex τ in the graph $\overline{G} = (V, \overline{A})$.

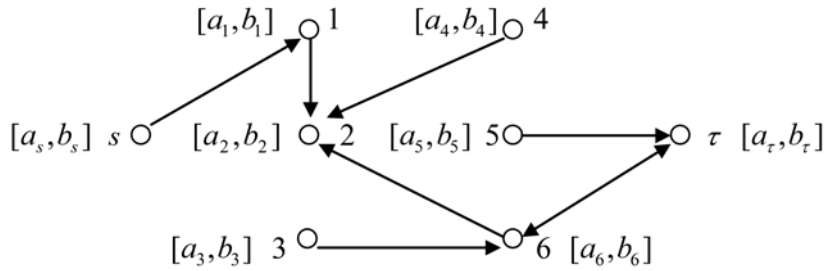

Figure 3. A represented graph $\overline{G} = (V, \overline{A})$ with time-windows.

So, the solution of the IMFPTW has solution. The network $G^{//} = (V, A^{//}, u^{//}, \tau, s)$ is presented by Figure 4.

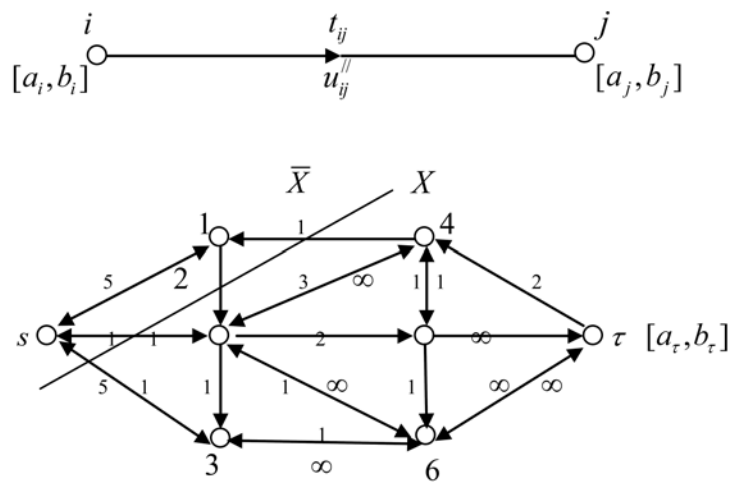

Figure 4. The network $G^{//} = (V, A^{//}, u^{//}, \tau, s)$ with time-windows.

The direct arcs of the minimum time-windows $\tau - s$ cut in the network $G^{//}$ is given by

$$B = (\{2, 3, 4, 5, 6, \tau\}, \{s, 1\}) = (X, \overline{X}) = \{(3, s), (2, s), (4, 1)\}, \qquad (22)$$

and $u^{//}(b) = 7$. This means that the lower bounds of a direct arcs of the $s - \tau$ cut time-windows is given by

$$(\overline{X},\ X) = \{(1,\ 4),\ (1,\ 2),\ (s,\ 2),\ (s,\ 3)\}, \tag{23}$$

and the upper bound of the only inverse arc $(2,\ s)$ of the $s - \tau$ cut in the time windows network G will be changed. The lower bounds of the arc $(1,\ 2)$ is already equal the flow on this arc. So it will be changed. The solution of the IMFPTW for the network G and the flow f is given by Figure 5.

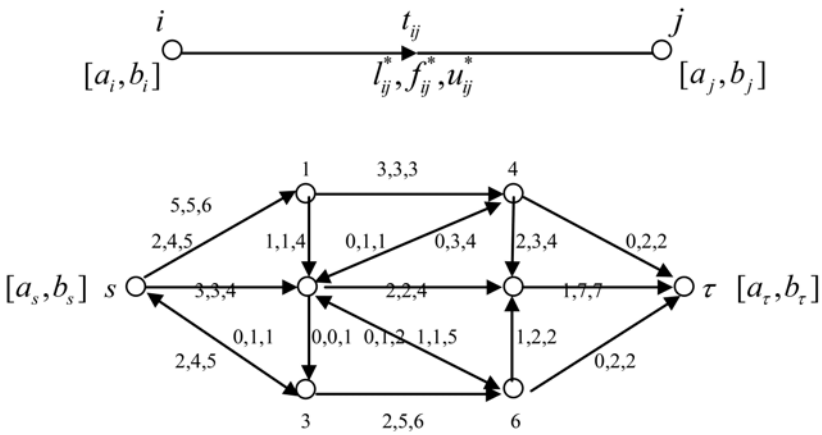

Figure 5. The network $G^* = (V,\ A,\ u_{ij}^*,\ l_{ij}^*,\ s,\ \tau)$ with time-windows.

6. Conclusion

We present a new algorithm of a new version of the inverse minimum flow problem. Strongly and weakly polynomial algorithms for solving the IMFPTW were proposed. The IMFPTW is reduced to the problem for finding efficiently the minimum time windows $\tau - s$ cut in the modified residual network $G^{//}$. It is possible from the beginning to decide fast, linear time and space complexity, if the IMFPTW has solution. The IMFPTW when only the lower bounds and the upper bounds for the flow can be changed are treated as particular cases of the IMFPTW. Also, an application instance is given.

References

[1] R. Ahuja, T. Magnanti and J. Orlin, Network Flows: Theory, Algorithms and Applications, Prentice Hall, Englewood Cliffs, NJ, 1993.

[2] R. Ahuja and J. Orlin, A faster algorithm for the inverse spanning tree problem, Journal of Algorithms 34 (2000), 177-193.

[3] D. Burton, B. Paulleyblank and Ph. Toint, The inverse shortest path problem with upper bounds on shortest paths costs, Network Optimization (P. Pardalos, D. Hearn and W. Harger), Lecture Notes in Economics, Math. Systems, Vol. 450, pp. 156-171, Springer-Verlang, Berlin/New York, 1997.

[4] D. Burton and Ph. Toint, On an instance of the inverse shortest path problem, Math. Programming 53 (1992), 45-61.

[5] M. Cai and X. Yang, Inverse Shortest Path Problems, Technical Report, Institute of Systems Sciences, Academia Sinica, Beijing, China, 1994.

[6] E. Ciurea and L. Ciupala, Sequential and parallel algorithms for minimum flows, Journal Appl. Math. & Computing 15(1-2) (2004), 53-75.

[7] N. El-Sherbeny, Difference of convex functions optimization methods: Algorithm of minimum-maximal network flow problem with time-windows, International Journal of Sciences: Basic and Applied Research 25(1) (2016), 67-81.

http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

[8] N. El-Sherbeny, The dynamic shortest path problems of minimum cost length time-windows and time-varying costs, International Journal of Scientific and Innovative Mathematical Research 3(3) (2015), 47-55.

www.arcjournal.org

[9] N. El-Sherbeny, The algorithm of the time-dependent shortest path problem with time-windows, Applied Mathematics 5(17) (2014), 2764-2770.

http://dx.doi.org/10.4236/am.2014.517264

[10] N. El-Sherbeny, Imprecision and flexible constraints in fuzzy vehicle routing problem, American Journal of Mathematical and Management Sciences 31 (2011), 55-71.

http://dx.doi.org/10.1080/01966324.2011.107378000

[11] N. El-Sherbeny, Minimum cost flow time-windows problem with interval bounds and flows, Theoretical Mathematics & Applications, Scienpress Ltd. 6 (2016), 15-32.

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

[13] A. Goldberg and S. Rao, Length Functions for Flow Computations, Technical Report, NEC Research Institute, Inc., 1997.

[14] A. Goldberg and R. Tarjan, A new approach to the maximum-flow problem, Journal of the Association for Computing Machinery 35 (1988), 921-949.

[15] D. Tuyttens, J. Teghem and N. El-Sherbeny, A Particular Multiobjective Vehicle Routing Problem Solved by Simulated Annealing, Lecture Notes in Economics and Mathematical Systems 535 (2004), 133-152, Springer-Verlag, Berlin, Germany.

http://dx.doi.org/10.1007/978-3-642-17144-4

[16] C. Yang, J. Zhang and Z. Ma, Inverse maximum flow and minimum cut problems, Optimization 40 (1998), 147-170.

■