Theoretical Mathematics & Applications, vol. 6, no. 3, 2016, 15-32
ISSN: 1792- 9687 (print), 1792-9709 (online)
Scienpress Ltd, 2016

Minimum Cost Flow Time-Windows Problem with Interval Bounds and Flows

Nasser A. El-Sherbeny[1]

Abstract

In this paper, we present and describe a new version of the Minimum Cost Flow Problem (MCFP). This version is the Minimum Cost Flow Time-Windows Problem with Interval Bounds and Flows (MCFTWPIBF). The MCFTWPIBF is a combinatorial optimization and an NP-hard problem. The minimum cost flow time-windows problem with interval data can be using two minimum cost flow time-windows problems with crisp data. In this paper, the idea of Ghiyasvand was extended the minimum cost flow time-windows problem with interval-valued lower, upper bounds and flows. Also, this work is extended to network with fuzzy lower, upper bounds and flows. A representation example network is given.

Mathematics Subject Classification: 90C27; 90C35; 68R10

Keywords: Optimization network flow; Minimum flow problem; Interval data; Time-windows

---

[1] Mathematics Department, Faculty of Science, Al-Azhar University, Nasr City-(11884), Cairo, Egypt.
Mathematics Department, Faculty of Applied Medical Science, Taif University, Turabah-(21995), KSA.
E-mail: nasserelsherbeny@yahoo.com

Article Info: *Received:* March 26, 2016. *Revised:* April 30, 2016.
Published online : July 30, 2016.

1 Introduction

Consider a directed network $G = (N, A)$ where N is a set of n nodes and A is a set of m arcs. We associate with each arc $(i, j) \in A$ an upper bound u_{ij} that denotes the maximum amount that can flow on the arc and a lower bound l_{ij} that denotes the minimum amount that must flow on the arc. Each arc has a non-negative transit time t_{ij}, $i \neq j$ and $i, j \in N$. Each node $i \in N$, has a time-window $[a_i, b_i]$, within which the node may be served, i.e., $a_i \leq t_i \leq b_i$, $t_i \in T$ is a non-negative service and leaving for that node. To define the minimum cost flow time-windows problem, we distinguish two special nodes in the network, namely a source node s and a sink node τ with time windows $[a_s, b_s]$ and $[a_\tau, b_\tau]$ respectively, see, El-Sherbeny [15], El-Sherbeny [16], El-Sherbeny [17], and Tuyttens, Teghem and El-Sherbeny [23]. The problem is to find the minimum cost flow with time-windows from the source node s to the sink node τ that satisfy the lower, upper bounds and balance constraints at all nodes. The decision variables in the minimum cost flow time-windows problem are arc flows, f_{ij} on an arc $(i, j) \in A$.

There are several approaches to solve the minimum flow problem. For decreasing path algorithms by Ciupala and Ciurea [4], Ciupala and Ciurea [5] and Ciupala and Ciurea [7], pre-flow algorithms by Ciupala [3], Ciurea [10], Ciupala and Ciurea [6], and Ciupala and Ciurea [8]. For minimax which consists of finding a maximum flow from the sink node to the source node in the residual network by Bang-jenson and Gutin [1], and Ciupala and Ciurea [5], using dynamic tree implementations by Ciupala and Ciurea [9]. Also, Ciurea, Georgescu and Marinescu [13], solved the minimum flow problem for bipartite networks. Ciurea and Deaconu [11], and Ciurea and Deaconu [12], solved the inverse minimum flow problem.

In Ghiyasvand [18], a new method to solve the minimum cost flow problem with interval data is presented. First, it solves a minimum cost flow problem with lower bounds, flows, and costs, second it, shows a minimum cost flow problem with upper bounds, flows, and costs. Then, the method combines these two solutions to form an interval solution. Ghiyasvand [18], also proved that is the interval solution is optimal for the minimum cost flow problem with interval bounds, flows, and costs.

Here, we extend their idea to present and describe the minimum cost flow time-windows problem with interval bounds and flows. We show that the minimum cost flow time-windows problem can be using the two minimum flow time-windows problems with crisp data.

The reminder of this work consists of five sections including Introduction. Section 2 presents the basic concepts, definitions and reviews of some results about crisp, time-windows, fuzzy time-windows, interval and fuzzy data which are used in the subsequent sections. In section 3, we presented, described the mathematical model of MCFTWPIBF and presented the relationship between the minimum cost flow time-windows problems with interval data and crisp data. In section 4, we presented the minimum cost flow time-windows problem with fuzzy data is described and given a representation network example. Finally, the conclusion is given in Section 5.

2 Basic Concepts and Definitions

2.1 Mathematical Models

Consider a directed network $G=(N,A),$ where N is a set of n nodes, A is a set of m arcs with a non-negative transit time t_{ij}, $i \neq j$ and $i, j \in N.$ For each node $i \in N,$ a time windows $[a_i, b_i]$ within which the node may be served and

$a_i \le t_i \le b_i$, $t_i \in T$ is a non-negative service and leaving time of the node i. A source node s and a sink node τ with time-windows $[a_s, b_s]$ and $[a_\tau, b_\tau]$ respectively. We also associate with each arc $(i, j) \in A$ an upper bound u_{ij} that denotes the maximum amount that can flow on the arc and a lower bound l_{ij} that denotes the minimum amount that must flow on the arc. The decision variables in the minimum cost flow time-windows problem are arc flows and we represent the flow on an arc $(i, j) \in A$ by f_{ij}. A minimum cost flow time-windows problem can state formally as follows:

$\min v$

subject to $$\sum_{\{j:(i,j)\in A\}} f_{ij} - \sum_{\{j:(j,i)\in A\}} f_{ji} = \begin{cases} v & ,i = s \\ -v, & i = \tau \\ 0 & ,\forall x \in N - \{s, \tau\} \end{cases} \qquad (1)$$

$t_i + t_{ij} \le t_j, \forall i, j \in N, t_i, t_{ij} \in T$, where $a_i \le t_i \le b_i$ and $a_j \le t_j \le b_j$ (2)

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

The minimum cost flow time-windows problem is one of the network flow that computes the minimum cost flow time-windows between two given nodes, called source and sink nodes.

Definition 2.1.1 *A time-windows constraint is defined by, for each node* $i, j \in N$ *then, a time windows* $[a_i, b_i]$ *and* $[a_j, b_j]$ *respectively. Each arc* $(i, j) \in A$ *has a non-negative transit time* t_{ij}, $i \ne j$ *and* $i, j \in N$, *where* $a_i \le t_i \le b_i, a_j \le t_j \le b_j$, $t_i, t_j \in T$, *see Figure 1.*

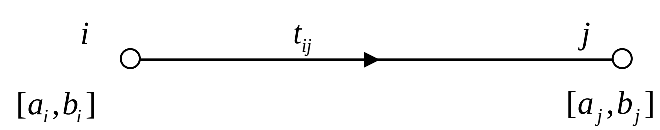


Figure 1: Node service with time-windows constraints

Let I denote the class of non-empty compact intervals $[\underline{x}, \overline{x}]$ on $[0, \infty)$. If $\overline{x} = \underline{x} = a,$ identity the interval with the real number $a.$

Definition 2.1.2 [20] *Let* $[\underline{x}_1, \overline{x}_1]$ *and* $[\underline{x}_2, \overline{x}_2]$ *be two compact intervals, then*

$$[\underline{x}_1, \overline{x}_1] + [\underline{x}_2, \overline{x}_2] = [\underline{x}_1 + \underline{x}_2, \overline{x}_1 + \overline{x}_2], \tag{4}$$

$$[\underline{x}_1, \overline{x}_1]\,[\underline{x}_2, \overline{x}_2] = [\min(\underline{x}_1\underline{x}_2, \underline{x}_1\overline{x}_2, \overline{x}_1\underline{x}_2, \overline{x}_1\overline{x}_2), \max\,(\underline{x}_1\underline{x}_2, \underline{x}_1\overline{x}_2, \overline{x}_1\underline{x}_2, \overline{x}_1\overline{x}_2)], \tag{5}$$

$$[\underline{x}_1, \overline{x}_1] \le [\underline{x}_2, \overline{x}_2] \text{ if } \underline{x}_1 \le \underline{x}_2 \text{ and } \overline{x}_1 \le \overline{x}_2\,. \tag{6}$$

The infimum and supremum of $[\underline{x}_1, \overline{x}_1]$ *and* $[\underline{x}_2, \overline{x}_2]$, *respectively, are defined by*

$$[\underline{x}_1, \overline{x}_1] \wedge [\underline{x}_2, \overline{x}_2] = [\min\{\underline{x}_1, \underline{x}_2\}, \min\{\overline{x}_1, \overline{x}_2\}] \tag{7}$$

$$[\underline{x}_1, \overline{x}_1] \vee [\underline{x}_2, \overline{x}_2] = [\max\{\underline{x}_1, \underline{x}_2\}, \max\{\overline{x}_1, \overline{x}_2\}] \tag{8}$$

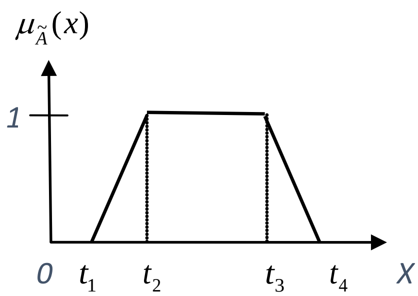
If $[\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n] \in I,$ then the infimum $\wedge_i\,[\underline{x}_i, \overline{x}_i]$ and supremum $\vee_i\,[\underline{x}_i, \overline{x}_i]$ are well-defined and

$$\sum_{\{i:i=1,2,\ldots,n\}} [\underline{x}_i, \overline{x}_i] = [\sum_{\{i:i=1,2,\ldots,n\}} \underline{x}_i, \sum_{\{i:i=1,2,\ldots,n\}} \overline{x}_i] \tag{9}$$

2.2 Fuzzy Time-Windows

Let $X = \Re^n$ be a non-empty set, $\tilde{A} \subset X$. The fuzzy set $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) : x \in X\}$ is the set of ordered pairs where $\mu_{\tilde{A}} : X \to [0,1]$ is the membership function of the fuzzy set $\tilde{A}$. The fuzzy constraint is a fuzzy set $\tilde{A} = (t_1, t_2, t_3, t_4)$ with flexible time-windows where (t_1, t_4) is the interval of non-zero satisfaction level and (t_2, t_3) is the interval of satisfaction level equal to 1 see, Figure 2.

The first step is to ask the expert to give a range for travel time between two places along with the most likely time; For example, the time $\tilde{T}$ to travel from point A to point B is between t_1 and t_3, but must possibly it is t_2. This sort of knowledge lets us construct 3-point fuzzy travel times see Figure 3.



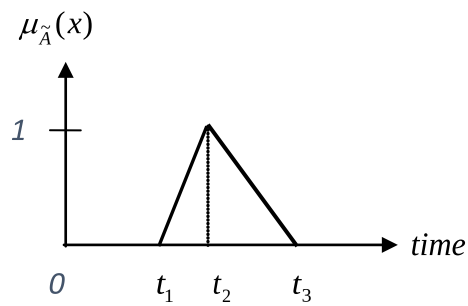
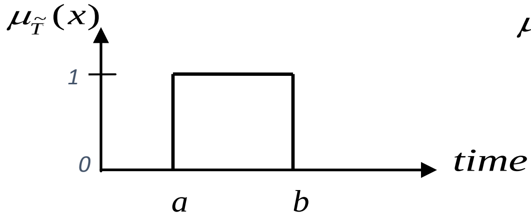


Figure 2: 4-Points representation of fuzzy interval Figure 3: Fuzzy travel time

Similarly obtain a fuzzy time-windows. Every node $i \in N$ is assigned by the expert to one of two predetermined groups; a classical fuzzy time-windows and fuzzy time-windows of a normal node. In an extreme case, fuzzy time-windows are tighter than the classical counterpartsee, Figure 4 and 5. The shown characteristics of fuzzy time-windows are suggested to the shipper who is allowed to modify them.



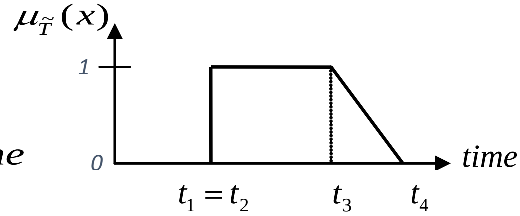


Figure 4: Classical fuzzy time-windows Figure 5: Fuzzy time-windows of a normal node

The $\beta-$level set of a fuzzy set $\tilde{A},$ denoted by $[\tilde{A}]^{\beta},$ is the crisp subset of X that contains all of elements with at least the given degree of membership $\beta.$

$$[\tilde{A}]^{\beta}=\{x\in X:\mu_{\tilde{A}}(x)\geq\beta\} \tag{10}$$

Fuzzy numbers are the class ξ^{1} of normal, upper semi-continuous fuzzy convex fuzzy sets on $\Re.$ That is, A is fuzzy number if all the level sets $[\tilde{A}]^{\beta}$, $0\leq\beta\leq 1,$ are compact intervals and there is at least one $z\in\Re$ such that $\mu_{\tilde{A}}(z)=1.$

Zadeh's extension principle, Hanss [19] is said to be one of the most important tools in fuzzy logic. It gives means to generalize non-fuzzy concepts, e.g., mathematical operations, to fuzzy sets. Let $\tilde{A}_1,\tilde{A}_2,...,\tilde{A}_n$ be fuzzy sets, defined on $X_1,X_2,...,X_n$, and f be a function $f:X_1\times X_2\times...\times X_n\rightarrow V.$ Zadeh's extension principle of f operating on $\tilde{A}_1,\tilde{A}_2,...,\tilde{A}_n$ gives a membership function, fuzzy set $\tilde{F}.$

$$\mu_{\tilde{F}}(v)=\sup_{\{u_1,u_2,...,u_n\in f^{-1}(v)\}}\min(\mu_{\tilde{A}_1}(u_1),...,\mu_{\tilde{A}_n}(u_n)), \tag{11}$$

where the inverse of f exist. Otherwise define $\mu_{\tilde{F}}(v)=0.$ Function f is called inducing mapping. If the domain is either discrete or compact, sup-min can be replaced by max-min.

- Basically, Zadeh's extension principle says that a fuzzy set is a collection of intervals with a membership associated ($\beta-$level) to them by Nguyen [21] and Nguyen and Kreinovich [22]. Thus, whenever a result can mention about the intervals, it can mention about the fuzzy sets.

Zadeh's extension principle is equivalent to the partition into β-cuts, using addition and scalar multiplication of convex sets, when the fuzzification uses Definition 2.1.2 for max-min and order operations.

- In Diamond [14], if a network flow theorem can be proved for compact interval-valued flows, capacities and costs, using Definition 2.1.2 for max, min and order properties, then a similar theorem will follow for fuzzy network quantities, which characterize a fuzzy number by its level sets, which are compact intervals.

2.3 Application of the Minimum Flow Problem

The famous application for the minimum flow problem is the machine set-up problem that is presented in this part: A job shop needs to perform p tasks on a particular day. The start and end times of each performance is given, the workers must perform these tasks according to the schedule so that exactly one worker perform each task. A worker can't work on two jobs at the same time. Also we have the set-up time required for a worker to go from one task to another. We wish to find the minimum number of workers to perform the task. This problem can be formulated as a minimum flow problem, Ciupala and Ciurea [4].

3 The Minimum Cost Flow Time-Windows Problem with Interval Data and Flows (MCFTWPIDF)

We presented and described the mathematical model of MCFTWPIBF and presented the relationship between the minimum cost flow time-windows problems with interval data and crisp data. Consider a directed network $G=(N,A)$ where N is a set of n nodes, A is a set of m arcs such that the time-windows, lower bound, upper bound, and flow of each arc are known to fall within specific ranges expressed as compact intervals $\overline{tw}, \bar{l}, \bar{u},$ and $\bar{f}$ respectively. Thus, for each arc $(i,j) \in A,$ we have

$$\begin{aligned} & tw_{ij} \in \overline{tw}_{ij} = [l_{tw}(ij), r_{tw}(ij)], \\ & l_{ij} \in \bar{l}_{ij} = [l_l(ij), r_l(ij)], \\ & u_{ij} \in \bar{u}_{ij} = [l_u(ij), r_u(ij)], \\ & f_{ij} \in \bar{f}_{ij} = [l_f(ij), r_f(ij)] \end{aligned} \tag{12}$$

where, $l_{tw}(ij), r_{tw}(ij), l_l(ij), r_l(ij), l_u(ij), r_u(ij), l_f(ij)$ and $r_f(ij)$ are non-negative real values. The minimum cost flow time-windows problem with compact interval-valued lower and upper bounds and flow can be state as follows:

$\min[v_l, v_r]$

subject to $$\sum_{\{j:(i,j)\in A\}} \bar{f}_{ij} - \sum_{\{j:(j,i)\in A\}} \bar{f}_{ji} = \begin{cases} [v_l, v_r], & i = s, \\ -[v_l, v_r], & i = \tau, \\ [0,0], & \forall i \in N - \{s, \tau\}, \end{cases} \tag{13}$$

$$\bar{t}_i + \bar{t}_{ij} \le \bar{t}_j,\ \bar{a}_i \le \bar{t}_i \le \bar{b}_i,\ \bar{t}_i, \bar{t}_{ij} \in \bar{T}, i \ne j,\ \forall i, j \in N, \tag{14}$$

$$\bar{l}_{tw}(ij) \le \overline{tw}_{ij} \le \bar{u}_{tw}(ij), \forall (i, j) \in A, \tag{15}$$

$$\bar{l}_{ij} \le \bar{f}_{ij} \le \bar{u}_{ij}, \forall (i, j) \in A. \tag{16}$$

We call this problem the interval-minimum cost flow time-windows problem. Let $\bar{f}^*$ be an answer of this problem. For each arc $(i, j) \in A,$ we define any element of the interval $\bar{f}^*_{ij}$ as an answer for the interval-minimum cost flow time-windows problem. By definition 2.1.2, conditions (13), (14), (15) and (16) can be written the following:

$$\Big[\sum_{\{j:(i,j)\in A\}} l_f(ij), \sum_{\{j:(i,j)\in A} r_f(ij)\Big] - \Big[\sum_{\{j:(j,i)\in A\}} l_f(ji), \sum_{\{j:(j,i)\in A]} r_f(ji)\Big] =$$

$$= \begin{cases} [v_l, v_r], & i = s, \\ -[v_l, v_r], & i = \tau, \\ [0,0], & \forall i \in N - \{s, \tau\}, \end{cases} \tag{17}$$

$$t_i + t_{ij} \le t_j,\ a_i \le t_i \le b_i,\ t_i, t_{ij} \in T, i \ne j,\ \forall i, j \in N, \tag{18}$$

$$l_{tw}(ij) \le tw_{ij} \le u_{tw}(ij), \forall (i, j) \in A, \tag{19}$$

$$l_l(ij) \le l_f(ij) \le l_u(ij), \forall (i,j) \in A, \tag{20}$$

$$r_l(ij) \le r_f(ij) \le r_u(ij), \forall (i,j) \in A. \tag{21}$$

There for, a flow $\bar{f}$ is feasible for the interval-minimum cost flow time-windows problem if it satisfies the conditions (17), (18), (19), (20) and (21). Thus, the interval-minimum cost flow time-windows problem can be written by the following:

$$\wedge [v_l, v_r] : f \text{ satisfies the conditions (17), (18), (19), (20) and (1).} \tag{22}$$

We define the $l-$minimum cost flow time-windows problem by the following:

$\min v_l$

subject to $$\sum_{\{j:(i,j)\in A\}} l_f(ij) - \sum_{\{j:(j,i)\in A\}} l_f(ji) = \begin{cases} v_l, & i = s, \\ -v_l, & i = \tau, \\ 0, & \forall i \in N - \{s,\tau\}, \end{cases} \tag{23}$$

$$t_i + t_{ij} \le t_j,\ a_i \le t_i \le b_i,\ t_i, t_{ij} \in T, i \ne j,\ \forall i,j \in N, \tag{24}$$

$$l_l(ij) \le l_f(ij) \le l_u(ij), \forall (i,j) \in A. \tag{25}$$

We also define the $r-$minimum cost flow time-windows problem by the following:

$\min v_r$

subject to $$\sum_{\{j:(i,j)\in A\}} r_f(ij) - \sum_{\{j:(j,i)\in A\}} r_f(ji) = \begin{cases} v_r, & i = s, \\ -v_r, & i = \tau, \\ 0, & \forall i \in N - \{s,\tau\}, \end{cases} \tag{26}$$

$$t_i + t_{ij} \le t_j,\ a_i \le t_i \le b_i,\ t_i, t_{ij} \in T, i \ne j,\ \forall i,j \in N, \tag{27}$$

$$r_l(ij) \le r_f(ij) \le r_u(ij), \forall (i,j) \in A. \tag{28}$$

The relationship among the $l-$minimum cost flow time-windows problem and interval minimum cost flow time-windows problem is shown by the next theorem.

Theorem 3.1 *Let $l_{f_1^*}$ (resp. $r_{f_2^*}$) is an optimal flow for the $l-$minimum cost flow time-windows problem (resp. $r-$minimum cost flow time-windows problem). Then $\bar{f}^* = [l_{f_1^*}, r_{f_2^*}]$ is an optimal flow for the interval-minimum cost flow time-windows problem.*

Proof. We first shows that the flow $\bar{f}^*$ is a feasible flow for the interval-minimum cost flow time-windows problem. By the feasibility of $l_{f_1^*}$ in the $l-$minimum cost flow time-windows problem, we get

$$l_l(ij) \le l_{f_1^*}(ij) \le l_u(ij), \forall (i,j) \in A, \tag{29}$$

$$\sum_{\{j:(i,j)\in A\}} l_{f_1^*}(ij) - \sum_{\{j:(j,i)\in A\}} l_{f_1^*}(ji) = 0, \forall i \in N - \{s,\tau\}, \tag{30}$$

By satisfying a time-windows constraint, $t_i + t_{ij} \le t_j$, $a_i \le t_i \le b_i$, $t_i, t_{ij} \in T, i \ne j, \forall i,j \in N$. In a same way, $r_{f_2^*}$ is a feasible flow for the $l-$ minimum cost flow time-windows problem, so we have

$$r_l(ij) \le r_{f_2^*}(ij) \le r_u(ij), \forall (i,j) \in A, \tag{31}$$

$$\sum_{\{j:(i,j)\in A\}} r_{f_2^*}(ij) - \sum_{\{j:(j,i)\in A\}} r_{f_2^*}(ji) = 0, \forall i \in N - \{s,\tau\}, \tag{32}$$

also, by satisfying a time-windows constraint. By (17), (30) and (32), $\bar{f}^*$ satisfies in (13) and by (18), (19), (20), (21) and (31), it satisfies in (16). Thus, $\bar{f}^*$ is a feasible flow for the interval-minimum cost flow time-windows problem. The flow v_1^* (resp. v_r^*) is optimal for the $l-$minimum cost flow time-windows problem (resp. the $r-$minimum cost flow time-windows problem), so by (22) and definition 2.1.1, we yield that $[v_1^*, v_r^*]$ is an optimal flow for the interval-minimum cost flow time-windows problem.

There for, by theorem 3.1, for solving the interval-minimum cost flow time-windows problem, it is enough that we solve the $l-$minimum cost flow time-windows problems, which yields the following theorem.

Theorem 3.2 *The minimum cost flow time-windows problem with interval-valued lower bound, upper bounds and flows is solved using two minimum cost flow time-windows problem with crisp data.*

4 The minimum cost flow time-windows problem according with Zadeh's extension principle

In this section, the minimum cost flow time-windows problem with fuzzy lower, upper bounds and flows is solved using Theorem 3.2. Consider a directed network $G=(N,A)$ where N is a set of n nodes, A is a set of m arcs with a fuzzy time-windows, fuzzy lower, upper bounds, and flows $t\tilde{w}, \tilde{l}, \tilde{u}$ and $\tilde{f}$, respectively. We call the minimum cost flow time-windows problem with fuzzy data as the fuzzy minimum cost flow time-windowsproblem. As it was mentioned in the above of the application of the minimum flow problem, the interval representation of the $\beta-$level allows extending classical interval arithmetic to the case of fuzzy numbers. Interval arithmetic can be directly applied to every $\beta-$level to obtain the resulting fuzzy set. For each $\beta \in [0,1]$ and each arc $(i,j) \in A$, we define the $\beta-$level sets corresponding to $t\tilde{w}, \tilde{l}, \tilde{u}$ and $\tilde{f}$ as follows:

$$[t\tilde{w}_{ij}]^{\beta} = t\tilde{w}_{ij}(\beta) = [tw_l(ij,\beta), tw_r(ij,\beta)], \tag{33}$$

$$[\tilde{l}_{ij}]^{\beta} = \tilde{l}_{ij}(\beta) = [l_l(ij,\beta), r_l(ij,\beta)], \tag{34}$$

$$[\tilde{u}_{ij}]^{\beta} = \tilde{u}_{ij}(\beta) = [l_u(ij,\beta), r_u(ij,\beta)], \text{ and} \tag{35}$$

$$[\tilde{f}_{ij}]^{\beta} = \tilde{f}_{ij}(\beta) = [l_f(ij,\beta), r_f(ij,\beta)]. \tag{36}$$

The minimum cost flow time-windows problem with compact interval-valued lower, upper bounds, and flows given by the following:

$\min[v_l(\beta), v_r(\beta)]$

subject to $[\sum_{\{j:(i,j)\in A\}} l_f(ij,\beta), \sum_{\{j:(i,j)\in A\}} r_f(ij,\beta)] - [\sum_{\{j:(j,i)\in A\}} l_f(ji,\beta), \sum_{\{j:(j,i)\in A\}} r_f(ji,\beta)] =$

$$= \begin{cases} [v_l(\beta), v_r(\beta)], & i = s, \\ -[v_l(\beta), v_r(\beta)], & i = \tau, \\ [0,0], & \forall i \in N - \{s,\tau\}. \end{cases} \tag{37}$$

$$tw_l(ij,\beta) \le tw_f(ij,\beta) \le tw_u(ij,\beta), \forall (i,j) \in A, \tag{38}$$

$$l_l(ij,\beta) \le l_f(ij,\beta) \le l_u(ij,\beta), \forall (i,j) \in A, \tag{39}$$

$$r_l(ij,\beta) \le r_f(ij,\beta) \le r_u(ij,\beta), \forall (i,j) \in A. \tag{40}$$

We call the interval-valued time-windows network with data (33), (34), (35) and (36) as the $\beta -$interval minimum flow time-windows network. The interval flow $[\tilde{f}]^{\beta}$ is feasible in the (G,β) network if it satisfies in (37), (38), (39) and (40). There for $\tilde{f}$ is a feasible flow for the fuzzy-minimum flow time-windows problem if, at each $\beta -$level, $[\tilde{f}]^{\beta}$ is a feasible flow in the $\beta -$interval minimum flow time-windows problem. At each $\beta -$level, we define the $\beta -$interval minimum flow time-windows problem by the following:

$\min[v_l(\beta), v_r(\beta)]$

subject to $\bar{f}(.,\beta)$ satisfies in (4.5), (4.6), (4.7) and (4.8). (41)

Hence, for each $\beta \in [0,1]$, an interval-valued minimum flow $\bar{f}^*(ij,\beta) = [l_{f^*}(ij,\beta), r_{f^*}(ij,\beta)]$, for each arc $(i,j) \in A$, is found by solving the $\beta -$interval minimum flow time-windows problem. By Theorem 3.2, $l_{f^*}(ij,\beta)$'s and $r_{f^*}(ij,\beta)$'s are computed using $l-\beta-$interval and $r-\beta-$interval minimum cost flow time-windows problems defined by the following:

- The $l-\beta-$interval minimum cost flow time-windows problem:

$\min v_l(\beta)$

subject to $$\sum_{\{j:(i,j)\in A\}} l_f(ij,\beta) - \sum_{\{j:(j,i)\in A\}} l_f(ji,\beta) = \begin{cases} v_l(\beta), & i = s, \\ -v_l(\beta), & i = \tau, \\ 0, & \forall i \in N - \{s,\tau\}, \end{cases} \quad (42)$$

$$tw_l(ij,\beta) \le tw_f(ij,\beta) \le tw_u(ij,\beta), \forall (i,j) \in A, \quad (43)$$

$$l_l(ij,\beta) \le l_f(ij,\beta) \le l_u(ij,\beta), \forall (i,j) \in A. \quad (44)$$

- The $r-\beta-$interval minimum cost flow time-windows problem:

$\min v_r(\beta)$

subject to $$\sum_{\{j:(i,j)\in A\}} r_f(ij,\beta) - \sum_{\{j:(j,i)\in A\}} r_f(ji,\beta) = \begin{cases} v_r(\beta), & i = s, \\ -v_r(\beta), & i = \tau, \\ 0, & \forall i \in N - \{s,\tau\}, \end{cases} \quad (45)$$

$$tw_l(ij,\beta) \le tw_f(ij,\beta) \le tw_u(ij,\beta), \forall (i,j) \in A, \quad (46)$$

$$r_l(ij,\beta) \le r_f(ij,\beta) \le r_u(ij,\beta), \forall (i,j) \in A. \quad (47)$$

Since the representation of the $\beta-$levels are used instead of the fuzzy numbers, by Zadeh's extension, the result is accorded with Zadeh's extension principle. In general, any function of k intervals $F(\tilde{A}_1,\tilde{A}_2,...,\tilde{A}_k)$ of k intervals $\tilde{A}_1,\tilde{A}_2,...,\tilde{A}_k$ can be extended to fuzzy by defining $[F(\tilde{A}_1,\tilde{A}_2,...,\tilde{A}_k)]^\beta =$ $F([\tilde{A}_1]^\beta,[\tilde{A}_2]^\beta,...,[\tilde{A}_k]^\beta)$.

However, unless F preserves inclusion, in order to get a fuzzy number as the result, we must modify the definition so that the level set β is a subset of $\beta \succ \beta'$. There for we define by Bondia, Sala and Sainz [2]:

$$[F(\tilde{A}_1,\tilde{A}_2,...,\tilde{A}_k)]^\beta = \cap_{0\le\beta'\le\beta} F([\tilde{A}_1]^{\beta'},[\tilde{A}_2]^{\beta'},...,[\tilde{A}_k]^{\beta'}). \quad (48)$$

For $\beta \in [0,1]$ and its (G,β) problem, consider an interval-valued minimum cost flow time-windows $\bar{f}^*(ij,\beta) = [l_{f^*}(ij,\beta), r_{f^*}(ij,\beta)]$, for each arc $(i,j) \in A$. Let $z^*(\beta) = \sum_{(i,j)\in A} \bar{c}(ij,\beta)\,\bar{f}^*(ij,\beta)$.

- **Representation example network:**
  - The representation example network with fuzzy time-widows, fuzzy bounds and flows, for a given β:

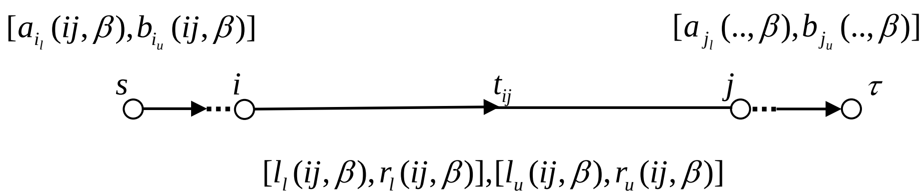


Figure 6: A network with fuzzy time-windows, fuzzy bounds and flows

  - The representation example network corresponding to the $l-\beta$ interval minimum cost flow time-windows problem:

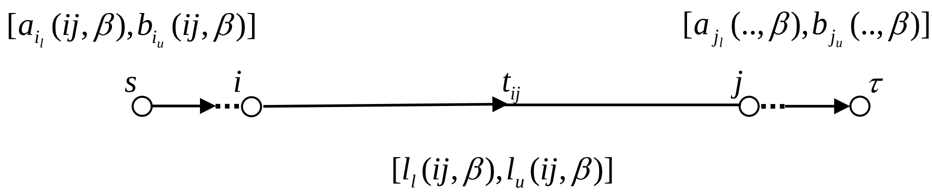


Figure 7: A network of $l-\beta$ interval minimum cost flow time-windows problem

  - The representation example network corresponding $r-\beta$ interval minimum cost flow time-windows problem:

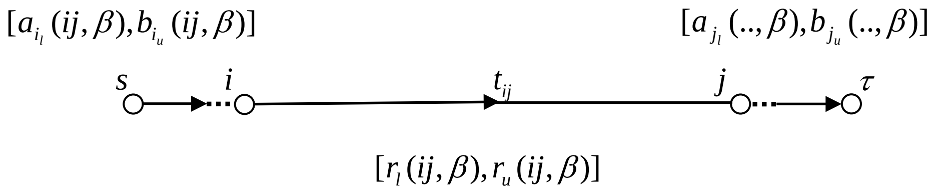


Figure 8: A network of the $r-\beta$ interval minimum cost flow time-windows problem

5 Conclusion

In this paper, we present and described a new version of the Minimum Cost Flow Problem (MCFP), a new version is a MCFTWPIBF. Ghiyasvand [18], presented a method to solve the minimum cost flow problem with interval date, which solves the problem using two minimum cost flow problems with crisp data. This paper extended the method of Ghiyasvand [18], by using the two minimum cost flow time-windows problems with crisp data. Also, this method is extended to the minimum cost flow problem with fuzzy time-windows, fuzzy lower, upper bounds and flow. A representation example network is given.

ACKNOWLEDGEMENTS. The author would like to thank a referee for his helpful comments and carful readings.

References

[1] Bang-jenson, J. and Gutin, G., *Digraphs: Theory, algorithms and applications*, London: Springer-Verlag, 2001.

[2] Bondia, J., Sala, A., Pico, J. and Sainz, A., Controller design under fuzzy pole-placement specifications: an interval arithmetic approach, *IEEE T. Fuzzy Systems*, **14**, (2006), 822-836.

[3] Ciupala, L., A deficit scaling algorithm for the minimum flow problem, *Sadhana*, **31**, (2006), 1169-1174.

[4] Ciupala, L. and Ciurea, E., Algorithm for the minimum flows, *Computer Science Journal of Moldova*, **9**, (2001a), 275-290.

[5] Ciupala, L. and Ciurea, E., An approach of the minimum flow problem, *The F^{ith} International Symposium of Economic Informatics*, (2001b), 786-790.

[6] Ciupala, L. and Ciurea, E., An algorithm for the minimum flow problem, *The Sixth International Conference of Economic Information*, (2003), 167-170.

[7] Ciupala, L. and Ciurea, E., Sequential and parallel algorithm for minimum flows, *J. Appl. Math. and Comput.*, **15**, (2004), 53-78.

[8] Ciupala, L. and Ciurea, E., A highest-label preflow algorithm for the minimum flow problem, *Proceedings of 11th WSEAS International Conference on Computers*, (2007), 26-28.

[9] Ciupala, L. and Ciurea, E., About perflow algorithms for the minimum flow problem, *WSEAS Transactions on computer Research*, **3**, (2008), 35-41.

[10] Ciurea, E., The wave perflow algorithm for the minimum flow problem, *Proceeding of the 10th WSAES International Conference on Mathematical and Computational Methods in Science and Engineering*, (2008), 473-476.

[11] Ciurea, E. and Deaconu, A., Inverse minimum flow problem, *J. Apple. Math. and Comput.*, **23**, 2007, 193-203.

[12] Ciurea, E. and Deaconu, A., Minimum flows in bipartite networks with unit capacities, *Proceedings of the 13th WSEAS International Conference on Computer*, (2009), 313-317.

[13] Ciurea, E., Georgescu, O. and Marinescu, D., Improved algorithm for minimum flows in bipartite network, *Int. J. Comput.*, **4**, (2008b), 351-360.

[14] Diamond, M., A fuzzy max-flow min-cut theorem, *Fuzzy Sets and Systems*, **119**, (2001), 139-148.

[15] El-Sherbeny, N., Minimum Convex and Differentiable Cost Flow Problem with Time-Window, *International Journal of Sciences: Basic and Applied Research*, **20**, (1), (2015), 139-150. http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

[16] El-Sherbeny, N., The Algorithm of the Time-Dependent Shortest Path Problem with Time-Window, *Applied Mathematics*, **5**(17), (2014), 2764-2770. http://dx.doi.org/10.4236/am.2014.517264

[17] El-Sherbeny, N., Imprecision and flexible constraints in fuzzy vehicle routing problem, *American Journal of Mathematical and Management Sciences*, **31**, (2011), 55-71. http://dx.doi.org/10.1080/01966324.2011.10737800

[18] Ghiyasvand, M., A new approach for solving the minimum cost flow problem with interval and fuzzy date, *International Journal of Uncertainty Fuzziness and Knowledge-Based Systems*, **19**, (2011), 71-88.

[19] Hanss, M., Applied fuzzy arithmetic, *An introduction with engineering applications*, Berlin, springer, 2005.

[20] Moore, E., *Methods and applications of interval analysis*, Philadelphia: SIAM, 1978.

[21] Nguyen, T., A note on the extension principle for fuzzy sets, *J. Math. Anal. and Appl.*, **64**, (1978), 359-380.

[22] Nguyen, T. and Kreinovich, V., Nested intervals and sets: Concepts, relations to fuzzy sets, and applications, In: R. Kearfott and V. Kreinovich (eds), *Application of Interval Computations*. Dordrecht: Kluwer, (1996), 245-290.

[23] Tuyttens, D., Teghem, J. and El-Sherbeny, N., 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