Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 14, Number 2 (2018), pp. 219-231
© Research India Publications
http://www.ripublication.com

# Algorithm of Fuzzy Minimum Cost Flow Problem with Fuzzy Time-Windows

**Nasser A. El-Sherbeny**

*Mathematics Department, Faculty of Science, Al-Azhar University,*
*Nasr City (11884), Cairo, Egypt.*
*Mathematics Department, Faculty of Applied Medical Science, Taif University, KSA.*

**Abstract**

In real world problems, optimization techniques of a network flow are useful for solving problems like, minimum flow problems, maximum flow problems and assignment problems. The Minimum Flow Problem (MFP) is one of the classic combinatorial optimization and an NP-hard problem with many applications in computer networks and logistic networks. The Fuzzy Minimum Cost Flow Problem with Fuzzy Time Windows (FMCFPFTW) is an extension of the MFP, also we consider a generalized fuzzy version of the MFP. Finally, we presents a new algorithm of the FMCFPFTW with the efficient polynomial time algorithm for solving the FMCFPFTW. Our algorithm is basically based on Dijkstra's algorithm of successive divisions the capacities by multiples of two, it solves the FMCFPFTW as a sequence of $O(n^2)$ shortest path problems on the residual networks with runs in $O(n^2 m\theta)$ time.

**Key words:** Combinatorial optimization, Network flow, Minimum flow problem, Fuzzy time-windows.

**Mathematics Subject Classification:** 05C35, 68R10, 90C27.

## 1. INTRODUCTION

Network flow are of fundamental importance in computer science, communication networks, industrial engineering and many other areas. The minimum flow problem is one of the classic problems of network optimization and an NP-hard problem. The textbook [1] is an exhaustive reference on the subject, like shortest path problem and maximum flow problem see, [9, 10, 11] minimum cost flow problem is a central problem in network flow. Consequently, the minimum cost flow problem has been studied extensively in the literature see, [3, 14, 15].

Different approaches have been proposed to solve the minimum cost flow problem. The classical algorithm for solving the minimum cost flow problem see, [10] which is essentially a primal method and runs in exponential time in the worst case. There are numbers of different polynomial time algorithms for solving the minimum cost flow problem such as; the capacity-scaling approach of [4] with running time $O((m \log \beta)(m + n \log n))$ on networks with $n$ vertices, $m$ arcs and the maximum arc capacity $\beta$. The cost-scaling approach of [12] which runs in $O(nm \log n^2 / m \log nC)$ time in networks with maximum arc cost magnitude $C$, the double scaling of both costs and capacities of [2] which solves the problem in $O(nm \log \log \beta \log nC)$ time in the worst case. The strongly polynomial time of [17] with running time bounded by $O((m \log n)(m + n \log n))$ and the primal network simplex polynomial time of [16] which runs in $O(\min(n^2 m \log nC, n^2 m^2 \log n))$ time at most. The first four of these algorithms have idea is common, that is scaling or successive approximation. Scaling algorithms work by solving a sequence of a sub-problems which numeric parameters are more and more closely approximate those of the original problem. A solution for one sub-problem helps to solve the next sub-problem in the sequence.

We consider $G = (V, E, s, \tau)$ be a directed graph on $V$, where $V$ is a limited set of vertices, $E$ is the set of arcs which is a family of subset of $V$, $s$ is the source vertex and $\tau$ is the sink vertex. Each arc has a nonnegative transit time $t_{v_i, v_j}$, $i \neq j$; $i, j = 1, 2, ..., n$. For each vertex $v_i \in V$, has a time-windows $[a_{v_i}, b_{v_i}]$ within which the vertex may be served and $t_{v_i} \in [a_{v_i}, b_{v_i}]$, $t_{v_i} \in \Re^+$ is a nonnegative service and leaving time of the vertex $v_i$. A source vertex $s$, with time-windows $[a_s, b_s]$, a sink vertex $\tau$ with time-windows $[a_\tau, b_\tau]$ and $t_s$ is a departure time of the source vertex see, ([5], [6], [7], [8], [19]).

The rest of the paper is organized as follows: Section 2 presents the problem formulation with basic concepts and some definitions. In Section 3, we presents a time-windows and a fuzzy time-windows. In Section 4, we will present a new version of the Minimum Flow Problem (MFP), a new version is a Fuzzy Minimum Cost Flow Problem with Fuzzy Time-Windows (FMCFPFTW), and we proposed the mathematical formulation model of the FMCFPFTW. In Section 5, we will present a new algorithm of the FMCFPFTW based on Dijkstra's algorithm with the successive divisions of capacities by multiples of two, it solves the FMCFPFTW as a sequence of $O(n^2)$ shortest path problems on the residual networks with $n$ vertices, $m$ arcs and runs in $O(n^2 m \theta)$ time, where $\theta$ is the smallest integer greater than or equal to $\log \beta$, $\beta$ is the largest arc capacity of the network. Finally, the conclusion is given in Section 6.

## 2. BASIC CONCEPTS AND DEFINITIONS

We consider $G=(V,E,s,\tau)$ be a directed graph on $V,|V|=n,$ where $V$ is a limited set of vertices $v_i, i=1,2,...,n.$ $E$ is the set of arcs, $|E|=m,$ which is a family of subset of $V,$ $s$ is the source vertex and $\tau$ is the sink vertex. Each arc has a nonnegative integral capacity $b_{v_i,v_j}$ and a nonnegative transit time $t_{v_i,v_j},$ $i\neq j;\ i,j=1,2,...,n.$ For each vertex $v_i\in V,$ has a time-windows $[a_{v_i},b_{v_i}]$ within which the vertex may be served and $t_{v_i}\in[a_{v_i},b_{v_i}],\ t_{v_i}\in T\in\Re^+$ is a nonnegative service and leaving time of the vertex $v_i.$ A source vertex $s,$ with time-windows $[a_s,b_s],$ a sink vertex $\tau$ with time-windows $[a_\tau,b_\tau]$ and $t_s$ is a departure time of the source vertex. Each arc $e_{v_i,v_j}=(v_i,v_j)\in E$ has a nonnegative integral unit cost $c_{v_i,v_j}.$ The arc $e_{v_i,v_j}\in E$ is said to emanate from vertex $v_i,$ if the arc $e_{v_i,v_j}$ is an outgoing arc of vertex $v_i$ and incoming arc of vertex $v_j.$ The degree of a vertex is the number of incoming and outgoing arcs of the vertex.

**Definition 2.1** A labeling function $\alpha$ is defined as a function from a set of vertices $V$ to a set real numbers $\Re$, i.e., $\alpha: V\to\Re.$ A flow $f$ and a labeling function $\alpha$ are called compatible if the flow $f$ and labeling function $\alpha$ together satisfy the following conditions:

- If $\bar{c}_{v_i,v_j}=c_{v_i,v_j}-(\alpha_{v_j}-\alpha_{v_i})\succ 0,$ then $f_{v_i,v_j}=0,\forall(v_i,v_j)\in E,$ (1)
- If $\bar{c}_{v_i,v_j}=c_{v_i,v_j}-(\alpha_{v_j}-\alpha_{v_i})\prec 0,$ then $f_{v_i,v_j}=b_{v_i,v_j},\forall(v_i,v_j)\in E,$ (2)
- If $\bar{c}_{v_i,v_j}=c_{v_i,v_j}-(\alpha_{v_j}-\alpha_{v_i})=0,$ then $f_{v_i,v_j}\in[0,b_{v_i,v_j}],\forall(v_i,v_j)\in E,$ (3)

$\bar{c}_{v_i,v_j}=c_{v_i,v_j}-(\alpha_{v_j}-\alpha_{v_i})$ is called the reduced cost of the arc $(v_i,v_j)\in E.$

**Definition 2.2** A residual network $G(f)$ corresponding to a feasible flow $f$ is defined as follows; for the arc $(v_i,v_j)\in E$:

- If $f_{v_i,v_j}\prec b_{v_i,v_j},$ then the directed arc $(v_i,v_j)\in E$ has a flow $f_{v_i,v_j}\geq 0$ and a reduced cost $\bar{c}_{v_i,v_j}=c_{v_i,v_j}-(\alpha_{v_j}-\alpha_{v_i})\geq 0,$ (4)
- If $f_{v_i,v_j}=b_{v_i,v_j},$ then the arc $(v_i,v_j)\in E$ is ignored, (5)
- If $f_{v_i,v_j}\succ 0,$ then the reverse arc $(v_j,v_i)\in E$ has a flow $f_{v_j,v_i}=b_{v_i,v_j}-f_{v_i,v_j}\geq 0$ and a reduced cost $\bar{c}_{v_j,v_i}=-c_{v_i,v_j}+(\alpha_{v_j}-\alpha_{v_i})\geq 0,$ (6)

- If $f_{v_i,v_j} = 0,$ then the arc $(v_j, v_i) \in E$ is ignored. (7)

In a network flow, we introduce an additional arc $(\tau, s)$ which has cost $c_{\tau,s} = 0$ and the capacity $b_{\tau,s} = \infty.$

## 3. TIME-WINDOWS AND FUZZY TIME-WINDOWS

- **Time-windows**

**Definition 3.1** Let $G = (V, E, s, \tau)$ be a directed graph on $V,$ where $V$ is a limited set of vertices $v_i, i = 1,2,...,n,$ $E$ is the set of arcs, $s$ is the source vertex and $\tau$ is the sink vertex. A time-windows is defined by, for each vertex $v_i, v_j \in V$ has a time windows $[a_{v_i}, b_{v_i}]$ and $[a_{v_j}, b_{v_j}]$ respectively. Each arc $e_{v_i,v_j} = (v_i, v_j) \in E$ has a nonnegative integral largest capacity $\beta_{v_i,v_j},$ a non-negative integral unit fuzzy cost $c_{v_i,v_j}$ and a non-negative transit time $t_{v_i,v_j},$ $i \neq j;$ $i, j = 1,2,...,n$ see, Figure 1.

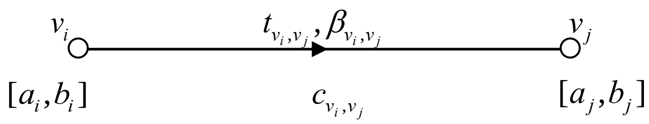


**Figure 1:** A representation time-windows of a nonnegative integer largest capacity arc.

- **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.

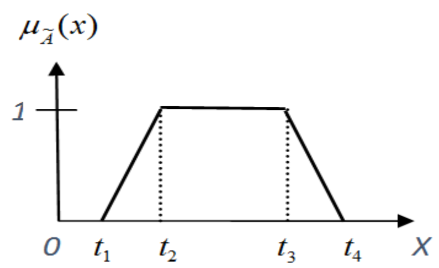


**Figure 2:** 4-Points representation of fuzzy interval

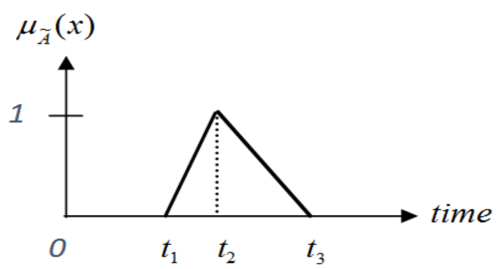


**Figure 3:** Fuzzy travel time

Similarly obtain fuzzy time-windows. Every vertex $v_i \in V, i = 1,2,...,n$ is assigned by the expert to one of two predetermined groups; a classical fuzzy time-windows and fuzzy time-windows of a normal vertex. In an extreme case, fuzzy time-windows are tighter than the classical counterpart see, Figure 4 and 5. The shown characteristics of fuzzy time-windows are suggested to the shipper who is allowed to modify them see, [9].

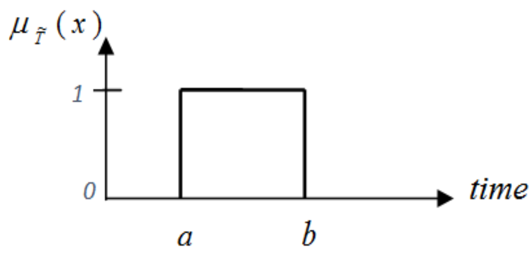


**Figure 4:** Classical fuzzy time-windows

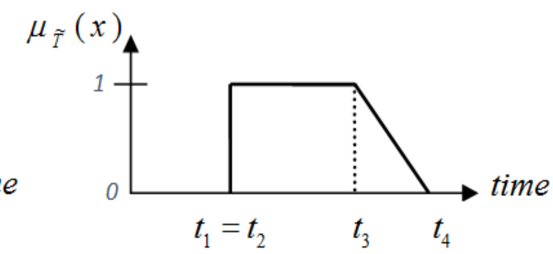


**Figure 5:** Fuzzy time-windows of a normal vertex

In many networks applications, there are exist the uncertain factors which can't expressed by fixed functions or parameters. Hence, the fuzzy theory is widely applied in networks see, ([13], [18], [20]). In this work, we consider the FMCFPFTW, the new optimization model is presented by virtue of fuzzy capacities calculating and crisp equivalents of fuzzy chance constraints.

## 4. MATHEMATICAL MODEL OF THE FMCFPFTW

Let $G = (V, E, s, \tau)$ be a directed graph on $V$, where $V$ is a limited set of vertices $v_i, i = 1,2,...,n,$ $E$ is the set of arcs, which is a family of subset of $V$, $s$ is the source vertex and $\tau$ is the sink vertex. A source vertex $s$, a sink vertex $\tau$ with fuzzy time-windows $[a_s, b_s], [a_\tau, b_\tau]$ respectively. Each arc has a nonnegative fuzzy capacity $b_{v_i,v_j}$ and a travel fuzzy time $t_{v_i,v_j}; i, j = 1,2,...,n.$ For each vertex $v_i \in V$, a fuzzy time-windows $[a_{v_i}, b_{v_i}]$ within which the vertex may be served and $t_{v_i} \in [a_{v_i}, b_{v_i}]$, $t_{v_i} \in \Re^+$ is a nonnegative service and leaving time of the vertex.

Let $f$ be a given feasible flow defined by a function $f : S \to \Re^{+}$ in a directed network $G$, a feasible flow has to satisfy the fuzzy capacity constraint, the fuzzy flow anti-symmetric constraint, the fuzzy flow conservation constraint and the fuzzy time-windows constraint of a directed network $G$.

The total fuzzy cost of a flow $\tilde{f}_{v_i,v_j}$ from the source vertex $s$ to the sink vertex $\tau$ with value $\tilde{z}$ is:

$$\sum_{(v_i,v_j)\in E} \tilde{c}_{v_i,v_j} \tilde{f}_{v_i,v_j} \tag{8}$$

The problem is to find a fuzzy maximum flow of a minimum fuzzy cost among the source vertex $s$ to the sink vertex $\tau$ which satisfy a fuzzy time windows constraint with value $\tilde{z}$. For each pair of vertices $(v_i, v_j) \in E$ with a fuzzy time-windows $[a_{v_i}, b_{v_i}]$, $[a_{v_j}, b_{v_j}]$ respectively, a fuzzy flow on arcs $\tilde{f}_{v_i,v_j}, (v_i, v_j) \in E, \quad i \neq j;$ $i, j = 1,2,...,n$ is satisfying the following conditions:

- The fuzzy capacity constraint,

  $$\tilde{f}_{v_i,v_j} \leq \tilde{b}_{v_i,v_j}, \forall (v_i, v_j) \in E,\ i \neq j;\ i, j = 1,2,...,n \tag{9}$$

- The fuzzy flow anti-symmetric constraint,

  $$\tilde{f}_{v_i,v_j} = -\tilde{f}_{v_j,v_i}, \forall (v_i, v_j) \in E,\ i \neq j;\ \ i, j = 1,2,...,n \tag{10}$$

- The fuzzy flow conservation constraint,

  $$\sum_{v_j \in V} \tilde{f}_{v_i,v_j} = 0, \forall v_i \in V \setminus \{s, \tau\}, i = 1,2,...,n \tag{11}$$

- The fuzzy time-windows constraint, for each vertex $v_i, v_j \in V$ has a fuzzy time windows $[a_{v_i}, b_{v_i}]$, $[a_{v_j}, b_{v_j}]$ respectively such that,

  $$a_{v_i} \leq t_{v_i} \leq b_{v_i},\ a_{v_j} \leq t_{v_j} \leq b_{v_j},\ t_{v_i} + t_{v_i,v_j} \leq t_{v_j};\ t_{v_i}, t_{v_j}, t_{v_i,v_j} \in \Re^{+}, i \neq j,\ \forall v_i, v_j \in V;$$
  $$i, j = 1,2,...,n$$
  (12)

We call a fuzzy flow $\tilde{f}_{v_i,v_j}$ extreme if it is a fuzzy minimum cost among flows with value $\tilde{f}_{s,\tau}$.

## 5. THE ALGORITHM FOR THE FUZZY MINIMUM COST FLOW PROBLEM WITH FUZZY TIME-WINDOWS

In this section, we presents a new algorithm of the FMCFPFTW. The FMCFPFTW can be find to a fuzzy minimum cost feasible flow with fuzzy time windows satisfies all the constraints in the networks with a polynomial time zero lower bound and $b_{v_i,v_j}$ upper bound on the fuzzy flow vector $f_{v_i,v_j}$ i.e. $0 \le f_{v_i,v_j} \le b_{v_i,v_j}, \forall (v_i,v_j) \in E, i \ne j;$ $i,j=1,2,...,n$ on the network $G=(V,E,s,\tau)$. Each vertex $v_i,v_j \in V$, has a fuzzy time-windows $[a_{v_i},b_{v_i}]$, $[a_{v_j},b_{v_j}]$ respectively, within which the vertex may be served and $t_{v_i} \in [a_{v_i},b_{v_i}], t_{v_j} \in [a_{v_j},b_{v_j}]$ is a nonnegative service and leaving time of the vertex with $t_{v_i}+t_{v_i,v_j} \le t_{v_j}$, $t_{v_i},t_{v_j},t_{v_i,v_j} \in \Re^+$, $\forall v_i,v_j \in V; i,j=1,2,...,n$ also, it is considered that $b_{v_i,v_j} \prec \infty$.

- <u>Initialization:</u>

Set $\vartheta = \min\{q \in Z^+ : 2^q \succ \max\{b_{v_i,v_j}, v_i, v_j \in V, i \ne j; i,j=1,2,...,n\}\}$

Set $\alpha_{v_i}=0$ for all vertices $v_i \in V, i=1,2,...,n$

Set $f_{v_i,v_j}=0$ and $\beta_{v_i,v_j}=b_{v_i,v_j}$ for all arcs $(v_i,v_j), i \ne j; i,j=1,2,...,n$

Set $f_{\tau,s}=0$ and $z=0$

- <u>Iteration:</u>

While $\vartheta \ge 1$, then do

Set $i=j=1, \vartheta=\vartheta-1$ and $z=2z$

Set $f_{v_i,v_j}=2f_{v_i,v_j}$ for all arcs $(v_i,v_j) \in E, a_{v_i} \le t_{v_i} \le b_{v_i}$ , $a_{v_j} \le t_{v_j} \le b_{v_j}$ and $t_{v_i}+t_{v_i,v_j} \le t_{v_j}, t_{v_i}, t_{v_j}, t_{v_i,v_j} \in \Re^+$, $i \ne j; i,j=1,2,...,n$

Set $f_{\tau,s}=2f_{\tau,s}$

Set $b_{v_i,v_j}=\left\lfloor \dfrac{\beta_{v_i,v_j}}{2^{\vartheta}} \right\rfloor$ for all arcs $(v_i,v_j) \in E, \forall v_i,v_j \in V$, $i \ne j; i,j=1,2,...,n$

While $v_i,v_j \in V; i,j \le n$, then do

If $f_{v_i,v_j} \le b_{v_i,v_j}$, then do

If $\bar{c}_{v_i,v_j}=c_{v_i,v_j}-(\alpha_{v_j}-\alpha_{v_i}) \prec 0$, then do

Do procedure $\Delta_{v_j,v_i}$ from $v_j$ to $v_i$ on the residual network $G(f_{v_i,v_j})$

If $v_i \in p,$ then do

Set $f_{v_i,v_j} = f_{v_i,v_j} + 1$

Set $f_{v_l,v_{l+1}} = f_{v_l,v_{l+1}} + 1$ for all $v_l$ on the shortest path $\mu$ of reduced costs from $v_j$ to $v_i$ in the network $G(f_{v_i,v_j})$

Set $f_{v_x,v_y} = f_{v_x,v_y} - 1$ for all backward arcs $(v_x, v_y)$ on the shortest path $\mu$

Set $z = z + \bar{c}_{v_i,v_j}$

End If

If $\bar{c}_{v_i,v_j} = c_{v_i,v_j} - (\alpha_{v_j} - \alpha_{v_i}) \prec 0$

Set $\alpha_{v_\zeta} = \alpha_{v_\zeta} - \bar{c}_{v_i,v_j}$ for all vertices $v_\zeta \in V, \zeta = 1,2,...,n;\ \alpha_{v_\zeta} \neq \alpha_{v_i}$

End If

End If

Do procedure $D(s,\tau)$ from $s$ to $\tau$ with a fuzzy time-windows $[a_s, b_s],\ [a_\tau, b_\tau]$ respectively on the new residual network $G(f)$

If $\tau \in p,$ then do

Set $f_{\tau,s} = f_{\tau,s} + 1$

Set $f_{v_x,v_y} = f_{v_x,v_y} + 1$ for all forward arcs $(v_x, v_y)$ on the shortest path $\mu$ of reduced costs from $s$ to $\tau$ in the network $G(f)$

Set $f_{v_x,v_y} = f_{v_x,v_y} - 1$ for all backward arcs $(v_x, v_y)$ on the shortest path $\mu$

Set $z = z + \alpha_\tau - \alpha_s$

End If

End If

Set $(v_i, v_j) = (v_i, v_j) + 1$

End While

End While

Set $\alpha_{v_i} = \alpha_{v_i} - \alpha_s, \forall i = 1,2,...,n$

End the algorithm

- Procedure: $\Delta(v_j^*, v_i^*)$

The procedure gives the shortest path of reduced Fuzzy cost between $v_i^*, v_j^* \in V$ where $[a_{v_i^*}, b_{v_i^*}]$, $[a_{v_j^*}, b_{v_j^*}]$ has a fuzzy time-windows of two vertices respectively, with $t_{v_i^*} \in [a_{v_i^*}, b_{v_i^*}], t_{v_j^*} \in [a_{v_j^*}, b_{v_j^*}]$, $t_{v_i^*} + t_{v_i^*, v_j^*} \le t_{v_j^*}$, $t_{v_i^*}, t_{v_j^*}, t_{v_i^*, v_j^*} \in \Re^+$; $i \ne j; i, j = 1,2,...,n$ on the defined residual network $G(f)$ based on Dijkstra's algorithm.

- Initialization:

Set $p = \phi,$

Set $I = \{1,2,...,n\},$

Set $v_j = 0,$

Set $d_{v_j} = \begin{cases} 0, v_j = v_j^* \\ \infty, v_j \ne v_j^* \end{cases}$ for all $v_j \in V, j = 1,2,...,n$

- Iteration:

While $I \ne \phi$ do

Let $h = \inf\{d_{v_i} : v_i \in I\}$

If $h = \infty$ do

Set $d_{v_i} = v_y, \forall v_i \in I$

Set $I = \phi$

Else do

Set $v_y = v_x$

Find $v_i \in I$ such that $d_{v_i} = v_y$

Set $I = I - \{v_i\}$ and $p = p \cup \{v_i\}$

For all $v_j \in I$ such that $(v_i, v_j) \in E$ is an arc in the residual network, do

If $d_{v_j} \succ v_y + \bar{c}_{v_i, v_j}$

Set pred $v_j = v_i$

End If

End For all

End If

End While

Set $\alpha_{v_i} = \alpha_{v_i} + d_{v_i}, \forall v_i \in V, i = 1,2,...,n$

End the procedure

- After the application of procedure $\Delta(v_j^*, v_i^*)$ on the defined residual network:

  - the new potential function $\alpha_{v_i}$ will be determined.

  - we found the set $p \neq \phi$ because $v_j^* \in p$ at least.

  - if $v_i^* \in p,$ then there is a path between $v_j^*, v_i^*$ on the defined residual network

The following procedure determines the shortest path of the reduced fuzzy costs $\mu$ defined by the vertices on the residual network from $v_j^*$ to $v_i^*$ in the case when there is a path between them.

**Identification of the shortest path from $v_j^*$ to $v_i^*$ on the defined residual network:**

- Initialization:

  While $v_i \neq v_j^*$ do

  Set $v_j =$ pred $v_i$

  Set $v_i = v_j$

  Set $\mu = v_i \cup \mu$

  End While

- **Remarks:**

1) When the algorithm is executed, the compatibility conditions between the current fuzzy flow $f_{v_i,v_j}$ and the current potential function $\alpha_{v_i,v_j}$ must be satisfied. In the contrary case, it is necessarily to change the current potential function by:

   For all arcs $(v_i, v_j) \in E, a_{v_i} \leq t_{v_i} \leq b_{v_i} \quad , a_{v_j} \leq t_{v_j} \leq b_{v_j}, \quad t_{v_i} + t_{v_i,v_j} \leq t_{v_j}$; $t_{v_i}, t_{v_i,v_j}, t_{v_j} \in \Re^+$, $i \neq j; i, j = 1,2,...,n$ with satisfies a fuzzy time windows, if $\bar{c}_{v_i,v_j} \leq 0,$ then we will change the vertex potentials to be:

$$\alpha_{v_\xi} = \alpha_{v_\xi} - \bar{c}_{v_i,v_j}, \forall v_\xi, v_i, v_j \in V, \xi = 1,2,...,n; v_\xi \neq v_j \tag{13}$$

2) The procedure $\Delta(v_j, v_i)$ is applied when $\bar{c}_{v_i,v_j} \leq 0,$ in this case, there are two possibilities:

- The first one, is $v_i \in p,$ i.e. there is a path $\mu$ from $v_j$ to $v_i$. In this case, it is found that $z^{**} \prec z^*,$ where $z^*, z^{**}$ is the old value and new value respectively, of the total fuzzy cost flow from a source vertex $s$ to a sink vertex $\tau,$ when the arc $(\tau, s) \in \mu.$

- The second one, is $v_i \notin p$ and in this case, $z$ does not change but the vertex potentials will be changed.

3) At the procedure $\Delta(s, \tau)$ is applied, there are two cases also possible:

- The first one is $\tau \in p,$ i.e. there is a path from a source $s$ to a sink $\tau,$ and it is found that $z^{**} \succ z^*,$ in this case.

- The second one is $\tau \notin p,$ and it is found that, $z$ does not change in this case but the potentials will be changed.

4) The new reduced fuzzy cost $\bar{c}^{**}_{v_i,v_j}$ is equal to the old reduced cost $\bar{c}^{*}_{v_i,v_j}$ minus the deference between $d_{v_j}$ and $d_{v_i}$ because, for all arcs $(v_i, v_j) \in E, a_{v_i} \leq t_{v_i} \leq b_{v_i}$ $, a_{v_j} \leq t_{v_j} \leq b_{v_j}$, $t_{v_i} + t_{v_i,v_j} \leq t_{v_j}; t_{v_i}, t_{v_j}, t_{v_i,v_j} \in \Re^+, i \neq j;$ $i, j = 1,2,...,n$ with $\alpha^*$ is old potential function then:

$$\begin{aligned}\bar{c}^{**}_{v_i,v_j} &= c_{v_i,v_j} - (\alpha_{v_j} - \alpha_{v_i}) = c_{v_i,v_j} - (\alpha^*_{v_j} + d_{v_j} - \alpha^*_{v_i} - d_{v_i}) = \\ &= c_{v_i,v_j} - (\alpha^*_{v_j} - \alpha^*_{v_i}) - (d_{v_j} - d_{v_i}) = \bar{c}^{*}_{v_i,v_j} - (d_{v_j} - d_{v_i})\end{aligned} \tag{14}$$

5) The reduced fuzzy cost $\bar{c}_{v_j,v_i}$ of the arc $(v_j, v_i) \in E$ is equal to the inverse reduced fuzzy cost $\bar{c}_{v_i,v_j}$ of the arc $(v_i, v_j) \in E$ because:

$$\begin{aligned}\bar{c}_{v_j,v_i} &= c_{v_j,v_i} - (\alpha_{v_i} - \alpha_{v_j}) = -c_{v_i,v_j} - (\alpha_{v_i} - \alpha_{v_j}) = -(c_{v_i,v_j} + (\alpha_{v_i} - \alpha_{v_j})) = \\ &= -(c_{v_i,v_j} - (\alpha_{v_j} - \alpha_{v_i})) = -\bar{c}_{v_i,v_j}\end{aligned} \tag{15}$$

The complexity of the algorithm time taken by the procedure $\Delta(v^*_j, v^*_i)$ which is based on Dijkstra's algorithm of $O(n^2)$ arithmetic operations with $n$ is the number of vertices in the network. The maximum number of iterations of the algorithm is $m\theta,$ where $m$ the number of arcs and $\theta$ is the smallest integer greater than or equal to $\log \beta,$ $\beta$ is the largest arc capacity of the network. The procedure $\Delta(v^*_j, v^*_i)$ is applied two times in each iteration, then the time taken by the algorithm at most $O(n^2 m\theta)$ arithmetic operations.

## 6. CONCLUSION

This paper presents a new version of the Minimum Flow Problem (MFP), a new version is the Fuzzy Minimum Cost Flow Problem with Fuzzy Time-Windows (FMCFPFTW) and a mathematical model is presented. We consider a generalized fuzzy version of the minimum flow problem. Also, we propose a new algorithm of the FMCFPFTW with the efficient polynomial time algorithm for solving the FMCFPFTW. Our algorithm is basically based on Dijkstra's algorithm of successive divisions the capacities by multiples of two. The result achieved in this work to illustrates the promising application prospects for algorithms.

## ACKNOWLEDGMENT

The author would like to thank an anonymous referee for reading carefully.

## REFERENCES

[1] R. Ahuja, T. Magnanti and J. Orlin, "Network flows: Theory, Algorithms and Applications", Prentice-Hall, Englewood Cliffs, NJ (1993).

[2] K. Ahuja, V. Goldberg, B. Orlin and E. Tarjan, "Finding minimum cost flows by double scaling", Mathematical Programming, 53 (1992), 243-266.

[3] G. Busacker and J.Gowen, "A procedure for determining a family of minimum-cost network flow patterns", Technical Report 15, Operation Research, Johns Hopkins University, 1961.

[4] J. Edmonds and R. Karp, "Theoretical improvements in algorithmic efficiency for network flow problems", Journal of ACM, (1972), 248-264.

[5] N. El-Sherbeny, "Resolution of a vehicle routing problem with multi-objective simulated annealing method", Ph.D., Mathematics Department, Faculty of Science, Mons University, Mons, Belgium, 2001.

[6] N. El-Sherbeny, "Algorithm of the inverse minimum flow problem with time-windows", Journal of Mathematical Science: Advances and Applications, 43 (2017), 1-18. http://dx.doi.org/10.18642/jmsaa_7100121736

[7] 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

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

[9] M. Ermolev and I. Mel'nik, "Extremal Problems on Graphs", (Russian), Kiev 1968.

[10] L. Ford and D. Fulkerson, "Rows in networks", Princeton University Press,

Princeton, New Jersey. 1962.

[11] L. Ford and D. Fulkerson, "Maximal flow through a network", Canadian Journal of Mathematics, 8 (1956), 399-404.

[12] V. Goldberg and E. Tarjan, "Solving minimum cost flow problem by successive approximation", Mathematics of Operation Research, 15 (1990), 430-466.

[13] T. Havens, J. Bezdek, and C. Leckie, "A soft modularity function for detecting fuzzy communities in social networks", IEEE Transactions on Fuzzy Systems, 21, 16 (2013), 1170-1175.

[14] S. Kapoor and M. Vaidya, "Fast algorithms for convex quadratic programming and multi-commodity flows", $18^{th}$ Annual Symposium on Theory of Computing, (1986), 147-159.

[15] T. Leighton and S. Rao, "An approximate max-flow min-cut theorem for uniform multi-commodity flow problems with applications to approximation algorithms", $29^{th}$ IEEE Annual Symposium on Foundations of Computer Science, (1988), 422-431.

[16] B. Orlin, "A polynomial time primal network simplex algorithm for minimum cost flows", Mathematical Programming, 78 (1997), 109-129.

[17] B. Orlin, "A faster strongly polynomial minimum cost flow algorithm", Operation Research, 41(2)(1993), 338-350.

[18] H. Turan, N. Kasap and H. Savran, "Using heuristic algorithms for capacity leasing and task allocation issues in telecommunication networks under fuzzy quality of service constraints", International Journal of Systems Science, 45, 3 (2014), 307-324.

[19] D. Tuyttens, J. Teghem, N. El-Sherbeny, "A particular multi-objective vehicle routing problem solved by simulated annealing", Lecture Notes in Economics and Mathematical Systems, Springer-Verlag, Berlin, Germany, 535 (2004), 133-152. http://dx.doi.org/10.1007/978-3-642-17144-4

[20] J. Wang, C. Wang and C. Chen, "The bounded capacity of fuzzy neural networks (FNNs) via a new fully connected neural fuzzy inference system (F-Confis) with its application", IEEE Transactions on Fuzzy Systems, 22, 6 (2014), 1373-1386.