Journal of Mathematical Sciences: Advances and Applications
Volume 55, 2019, Pages 1-20
Available at http://scientificadvances.co.in
DOI: http://dx.doi.org/10.18642/jmsaa_7100121950

AN ALGORITHM OF A MINIMUM COST FLOW PROBLEM WITH TIME-VARYING AND TIME-WINDOWS

NASSER A. EL-SHERBENY

Department of Mathematics
Faculty of Science
Al-Azhar University
Nasr City, 11884, Cairo
Egypt

Department of Mathematics
Turabah University College
Taif University
Turabah, 29541
Kingdom of Saudi Arabia
e-mail: nasserelsherbeny@yahoo.com

Abstract

The Minimum Cost Flow Problem (MCFP) is a logical and distribution problem which is one of the classical combinatorial optimization and an NP-hard problem. We propose a new version of a MCFP, this version is a Minimum Cost Flow Problem on a Time-Varying and Time-Windows (MCFPTVTW). A mathematical model of the MCFPTVTW is presented. The objective is to find an optimal schedule to send a flow from the source vertex s to it's the sink vertex ρ satisfies a time-varying and time-windows constraint with the minimum cost of

2010 Mathematics Subject Classification: 65G30, 05C35, 90C27, 68R10.
Keywords and phrases: minimum cost flow, combinatorial optimization, time-varying, time-windows.
Received April 2, 2018; Revised April 14, 2018

the arc $c_r(a, t_{\nu_i})$ and a minimum waiting times at vertices $c_w(\nu_i, t_{\nu_i})$, subject to the constraint that the flow must arrive at the sink vertex with time $t_{\nu_i} \leq T$. Finally, an algorithm of the MCFPTVTW is presented.

1. Introduction

The minimum cost flow problem (MCFP) is a basic problem in network flow theory with several applications, which is one of the classical combinatorial optimization and an NP-hard problem. The MCFP on a network has been studied extensively, see ([1, 2]). We consider $G = (V, A, l, b, c_r, c_w, [a_{\nu_i}, b_{\nu_i}], s, \rho)$ be a network without parallel arcs and loops, where V is a set of vertices and A is a set of arcs. For each vertex $\nu_i \in V$, $i = 1, 2, \ldots, n$ is an associated a three integer parameters, a waiting cost $c_w(\nu_i, t_{\nu_i})$, a vertex capacity $l(\nu_i, t_{\nu_i})$ and a time-windows $[a_{\nu_i}, b_{\nu_i}]$ For each arc $a = (\nu_i, \nu_{i+1}) \in A$ is an associated a three integer parameters, a positive transit time $b(a, t_{\nu_i})$, a transit cost $c_r(a, t_{\nu_i})$, and a positive capacity limit $l(a, t_{\nu_i})$. All these parameters are functions of the time $t_{\nu_i} \in [0, T]$, where $a_{\nu_i} \leq t_{\nu_i} \leq b_{\nu_i}$. The source vertex s and the sink vertex ρ are time-windows $[a_s, b_s]$, $[a_\rho, b_\rho]$ respectively, see ([8, 9, 10, 11, 16]), the flow must arrived at a vertex $\nu_i \in V$ before the time b_{ν_i}, if it arrives before a_{ν_i}, it must a waiting time a_{ν_i} before treating the flow at a vertex $\nu_i \in V$ (a waiting time $w(\nu_i) \succ 0$), see ([3, 4, 5, 6, 7]). The problem is to determine how given the amount of the flow can be sent from one vertex s (the source) to another vertex ρ (the sink) at the minimum cost, subject to the capacity limits on the arcs of a network, see ([12, 13, 14, 15). Traditionally, this problem is considered as a static one, where it is assumed that it takes zero time to traverse any arc, and all attributes of the network, including the cost to send a flow on the arc, and the capacity of the arc, are time invariant. A more realistic model is to take into account the time needed to traverse an arc. The transit times $b(\nu_i, \nu_{i+1})$ on the arc $(\nu_i, \nu_{i+1}) \in A$, $i = 1, 2, \ldots, n$

and time-windows $[a_{\nu_i}, b_{\nu_i}]$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$, $t_{\nu_i} \in [0, T]$ for each vertex $\nu_i \in V$, subject to the condition that the schedule remains optimal for any $t_{\nu_i} \le T$. The new version of the problem becomes so-called a minimum cost flow problem on a time-varying and time-windows (MCFPTVTW).

The remainder of this article is organized as follows, after the introduction; in Section 2, we will introduce some important basic concepts and definitions of a time-varying and time-windows of a shortest dynamic *f*-augmenting path and a dynamic residual network. In Section 3, we presented the dynamic residual network with a time-varying and time-windows. In Section 4, we propose, the general mathematical model of the MCFPTVTW. In Section 5, we consider an instance without a waiting time. In Section 6, we presented an algorithm of the MCFPTVTW. Finally, the conclusion is given in Section 7.

2. Basic Concepts, Notations and Definitions

Let $G = (V, A, l, b, c_r, c_w, [a_{\nu_i}, b_{\nu_i}], s, \rho)$ be a network without a parallel arcs and loops, where V is a set of vertices and A is a set of arcs. The parameters l, b, c_r, c_w of the network are functions of time $t_{\nu_i} \le T$, where $t_{\nu_i} \in [0, T]$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$, $i = 1, 2, \ldots, n$ is an integer number, $[a_{\nu_i}, b_{\nu_i}]$ is a time-windows for each vertex $\nu_i \in V$. More specifically, $l(\nu_i, \nu_{i+1}, t_{\nu_{i+1}})$ is the capacity of an arc $(\nu_i, \nu_{i+1}) \in A$ at time t_{ν_i}, $l(\nu_i, t_{\nu_i})$, $\nu_i \in V \setminus \{s, \rho\}$ is the capacity of a vertex ν_i during the period from t_{ν_i} to $t_{\nu_{i+1}}$, which is the maximum allowable flow that can wait at the vertex of any time during the period $[t_{\nu_i}, t_{\nu_{i+1}}]$, $b(\nu_i, \nu_{i+1}, t_{\nu_{i+1}})$ is the transit time needed to send a flow through the arc (ν_i, ν_{i+1}) at time $t_{\nu_{i+1}}$, $c_r(\nu_i, \nu_{i+1}, t_{\nu_{i+1}})$ is the cost for a transmitting one unit of flow through the arc (ν_i, ν_{i+1}) at time $t_{\nu_{i+1}}$ and

$c_w(\nu_i, t_{\nu_i})$ is the unit cost for keeping one unit of the flow waiting at the vertex ν_i for the duration $[t_{\nu_i}, t_{\nu_{i+1}}]$. We assume that b, l, c_r, and c_w are integers, and two particular vertices s and ρ are the source and the sink vertices, respectively.

In this article, we consider $n = |V|$, $m = |A|$, and $f(\nu_i, \nu_{i+1}, t_{\nu_{i+1}})$ be a flow value on the arc (ν_i, ν_{i+1}) during the period $[t_{\nu_i}, t_{\nu_i} + b(\nu_i, \nu_{i+1}, t_{\nu_{i+1}})]$, $f(\nu_i, t_{\nu_i})$, $\forall \nu_i \in V \setminus \{s, \rho\}$ is a flow value waiting at the vertex ν_i during the period $[t_{\nu_i}, t_{\nu_{i+1}}]$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$ and $f(\alpha, T)$ is the total flow value under a schedule α, which specifies when and how to send a flow from the source vertex s to the sink vertex ρ within a time-varying, time-windows and a time limit T, then

$$f(\alpha, T) = \sum_{(\nu_i, \rho) \in A, t_{\nu_i} + b(\nu_i, \rho, t_{\nu_i}) \le T} f(\nu_i, \rho, t_{\nu_i}). \tag{1}$$

A minimum cost flow problem with time-varying and time-windows is to find a feasible schedule to send a given flow ν_f from s to ρ within the time limit T so as to minimize the total cost satisfy all the constraints.

A time-varying and time-windows is an essential ingredient which we must consider in our model. Consequently, some important concepts and procedures will have to be re-defined and re-constructed, to take into account the existence of dynamic time. Without ambiguity, in the following, we will assume that the length of an arc is equal to its cost, and use interchangeably the terminologies cost and length of the shortest path and cheapest path. Further, we denote $P(s, \nu_i)$ as a directed path from s to ν_i connecting to the vertices $s = \nu_1, \nu_2, \ldots, \nu_n$; $\nu_i \in V$, $i = 1, 2, \ldots, n$, where $\beta(\nu_i)$, $w(\nu_i)$, and $\tau(\nu_i)$ are respectively, the arrival time, the waiting time, and the departure time at the vertex ν_i with a time t_{ν_i}, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$, $t_{\nu_i} \in [0, T]$. The path has a time t_{ν_i} if the total time required to traverse this path is $t_{\nu_i} \le T$.

Definition 2.1. Let $G = (V, A, l, b, c_r, c_w, [a_{\nu_i}, b_{\nu_i}], s, \tau)$ be a network where V is a limited set of vertices $\nu_i \in V$, $i = 1, 2, \ldots, n$ and A is a set of arcs. Each arc $(\nu_i, \nu_j) \in A$ has an integral numbers l, b, c_r, and c_w. The source vertex s and the sink vertex ρ with time-windows $[a_s, b_s]$, $[a_\rho, b_\rho]$, respectively. A time-window is defined by, for each vertex $\nu_i, \nu_j \in V$ has time-windows $[a_{\nu_i}, b_{\nu_i}]$ and $[a_{\nu_j}, b_{\nu_j}]$, respectively, $i \neq j$; $i, j = 1, 2, \ldots, n$, see Figure 1.

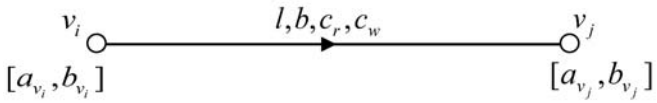

Figure 1. A representation of time-windows of the two vertices $\nu_i, \nu_j \in V$, $i \neq j$; $i, j = 1, 2, \ldots, n$.

Definition 2.2. The path $P(s, \nu_i)$ is said to be a dynamic f-augmenting path from s to ν_i with time-varying and time-windows $t_{\nu_i} \leq T$, $a_{\nu_i} \leq t_{\nu_i} \leq b_{\nu_i}$ if it satisfies

- $\tau(\nu_{i-1}) + b(\nu_{i-1}, \nu_i, \tau(\nu_{i-1})) = \beta(\nu_i)$, (2)
- $\beta(\nu_i) + w(\nu_i) = \tau(\nu_i)$, (3)
- $l(\nu_{i-1}, \nu_i, \tau(\nu_{i-1})) \succ 0$, (4)
- $l(\nu_{i-1}, t_{\nu_{i-1}}) \succ 0$, (5)

for $i = 2, 3, \ldots, n$ and $\tau(\nu_{i-1}) = \beta(\nu_{i-1})$, then there is no waiting time, i.e., $(w(\nu_{i-1}) = 0)$.

Definition 2.3. Let $P(s, \nu_i)$ be a dynamic f-augmenting path from s to ν_i, with time-varying and time-windows $t_{\nu_i} \le T$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$, the capacity of the path $P(s, \nu_i)$ is defined as $Cap(P(s, \nu_i)) = \min\{\delta_1, \delta_2\}$, where $\delta_1 = \min_{2\le i\le n} l(\nu_{i-1}, \nu_i, \tau(\nu_{i-1}))$ and $\delta_2 = \min_{2\le i\le n} \min_{0\le t_{\nu_i}\le w(\nu_{i-1})} l(\nu_{i-1}, \beta(\nu_{i-1}))$.

In this work, the algorithm to be developed will search, successively, the shortest paths from the source vertex s to the sink vertex ρ in a dynamic residual network and then the transmit time as much as possible flow along the paths, also, satisfy a time-varying and time-window constraint. In [13], we will need the network updating procedure to keep the needed information on the current dynamic flow, which is to be introduced below. In our network of the updating procedure, we create, initially, a new network to replace the original one.

Definition 2.4. For every arc $(\nu_i, \nu_{i+1}) \in A$, we create an artificial arc, denoted by $[\nu_{i+1}, \nu_i]$. It's a transit time $b[\nu_{i+1}, \nu_i, t_{\nu_i}]$, a transit cost $c_r[\nu_{i+1}, \nu_i, t_{\nu_i}]$, and the capacity $l[\nu_{i+1}, \nu_i, t_{\nu_i}]$, where $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$ are defined as follows:

For $\nu_i \in V$; $i = 1, 2, \ldots, n$

$$b[\nu_{i+1}, \nu_i, t_{\nu_i}] = -b(\nu_i, \nu_{i+1}, u) \text{ if } 0 \le t_{\nu_{i+1}} = u + b(\nu_i, \nu_{i+1}, u) \le T,\ 0 \le u \le T$$

$$= \infty \text{ otherwise,} \tag{6}$$

$$c_r[\nu_{i+1}, \nu_i, t_{\nu_i}] = -c_r(\nu_i, \nu_{i+1}, u) \text{ if } 0 \le t_{\nu_{i+1}} = u + b(\nu_i, \nu_{i+1}, u) \le T,\ 0 \le u \le T$$

$$= \infty \text{ otherwise,} \tag{7}$$

$$l[\nu_{i+1}, \nu_i, t_{\nu_i}] = 0,\ \forall(\nu_i, \nu_{i+1}) \in A,\ 0 \le t_{\nu_i} \le T,\ a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}. \tag{8}$$

For every vertex $\nu_i \in V$, we define $l[\nu_i, t_{\nu_i}]$ as the capacity within which a flow can be waiting at ν_i from time t_{ν_i} to $t_{\nu_{i-1}}$ and $c_w[\nu_i, t_{\nu_i}]$ as the cost from a flow to stay at ν_i from time t_{ν_i} to $t_{\nu_{i-1}}$. Initially, let

$$l[\nu_i, t_{\nu_i}] = 0 \text{ and } c_w[\nu_{i+1}, t_{\nu_{i+1}}] = -c_w(\nu_{i+1}, t_{\nu_{i+1}}), \forall i = 1, 2, \ldots, n; t_{\nu_i} \in [0, T]. \tag{9}$$

No flow can be send along any artificial arcs in the network as defined above since the capacities of these arcs are a set to zero. Hence, this a new network is equivalence to the original one, and so we will still denote it by G.

3. A Dynamic Residual Network with Time-Varying and Time-Windows

Let $P(s, \rho)$ be a dynamic f-augmenting path from s to ρ, and f_p be the flow value send along $P(s, \rho)$ which satisfies $0 \le f_p \le Cap(P(s, \rho))$. For $i = 1, 2, \ldots, n-1$; $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$, $t_{\nu_i} \in [0, T]$ and $[a_{\nu_i}, b_{\nu_i}]$ is a time-windows for each vertex $\nu_i \in V \setminus \{s, \rho\}$, then the update of the arc capacity is given by:

- If (ν_i, ν_{i+1}) is not an artificial arc. Let

$$l(\nu_i, \nu_{i+1}, \tau(\nu_i)) + f_p = l(\nu_i, \nu_{i+1}, \tau(\nu_i)), \tag{10}$$

$$l[\nu_{i+1}, \nu_i, \beta(\nu_{i+1})] - f_p = l[\nu_{i+1}, \nu_i, \beta(\nu_{i+1})]. \tag{11}$$

- If (ν_i, ν_{i+1}) is an artificial arc. Let

$$l[\nu_i, \nu_{i+1}, \tau(\nu_i)] + f_p = l(\nu_i, \nu_{i+1}, \tau(\nu_i)), \tag{12}$$

$$l(\nu_{i+1}, \nu_i, \beta(\nu_{i+1})) - f_p = l(\nu_{i+1}, \nu_i, \beta(\nu_{i+1})). \tag{13}$$

For $i = 1, 2, \ldots, n-1$, the update of a vertex capacity is given by:

- If the waiting time at vertex ν_i is positive $(w(\nu_i) \succ 0)$. Let

$$l(\nu_i, t_{\nu_i}) + f_p = l(\nu_i, t_{\nu_i}), \text{ where } t_{\nu_i} = \tau(\nu_i), \tag{14}$$

$$l[\nu_i, t_{\nu_i}] - f_p = l[\nu_i, t_{\nu_i}], \text{ where } t_{\nu_i} = \beta(\nu_i). \tag{15}$$

• If the waiting time at vertex ν_i is negative $(w(\nu_i) \prec 0)$. Let

$$l[\nu_i, t_{\nu_i}] + f_p = l[\nu_i, t_{\nu_i}], \text{ where } t_{\nu_i} = \tau(\nu_i), \tag{16}$$

$$l(\nu_i, t_{\nu_i}) - f_p = l(\nu_i, t_{\nu_i}), \text{ where } t_{\nu_i} = \beta(\nu_i). \tag{17}$$

The network generated by the network updating procedure above is said to be a dynamic residual network of the time-varying and time-windows.

The optimization problem in the original network and in a dynamic residual network time-windows is equivalent to the sense that there is a one-to-one correspondence between their feasible solutions.

In the original network, we assume that all transit times $b(\nu_i, \nu_{i+1}, t_{\nu_{i+1}}) \succ 0$. Thus, the first dynamic f-augmenting path will only contain arcs with positive a transit times $b(\nu_i, \nu_{i+1}, t_{\nu_{i+1}}) \succ 0$ and positive waiting times $w(\nu_i, t_{\nu_i}) \succ 0$. But in a dynamic residual network of the time-varying and time-windows, the transit time associated with an artificial arc is a negative number, and a flow can be stored at a vertex for a negative waiting time. Therefore, a dynamic f-augmenting path found in the dynamic residual network of a time-varying and time-windows may contain some arcs with negative a transit times and negative waiting times.

Definition 3.1. Let $\beta(\nu_i)$, $\tau(\nu_i)$ and $w(\nu_i)$, $i = 1, 2, \ldots, n$, $w(\nu_i) \prec 0$, then $P(s, \nu_i)$ is said to be f-augmenting path from s to ν_i. If for $i = 2, \ldots, n$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$, $t_{\nu_i} \le T$, then for each vertex $\nu_i \in V \setminus \{s, \rho\}$ it satisfies:

- $\tau(\nu_{i-1}) + b[\nu_{i-1}, \nu_i, \tau(\nu_{i-1})] = \beta(\nu_i)$, (18)
- $\beta(\nu_i) + w(\nu_i) = \tau(\nu_i)$, (19)
- $l[\nu_{i-1}, \nu_i, \tau(\nu_{i-1})] \succ 0$, if (ν_{i-1}, ν_i) is an artificial arc, (20)
- $l[\nu_{i-1}, t_{i-1}] \succ 0$, $t_{\nu_{i-1}} = \beta(\nu_{i-1})$, if $w(\nu_{i-1}) \prec 0$, where

  $0 \le \beta(\nu_i) \le T$, $0 \le \tau(\nu_i) \le T$, for $i = 2, \ldots, n$. (21)

Definition 3.2. For a vertex $\nu_i \in V \setminus \{s\}$ and a given time-varying and time-windows $t_{\nu_i} \le T$, where $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$, the cost of a shortest dynamic f-incrementing path from s to ν_i with time t_{ν_i} is said to be infinity if:

- there is no path from s to ν_i, or
- all paths from s to ν_i either have times greater than t_{ν_i} or a violate of some other constraints and thus the infeasible.

4. A Mathematical Model of the MCFPTVTW

- The general mathematical model of the problem is given by:

$$\min \sum_{(\nu_i, \nu_{i+1}) \in A} \sum_{t_{\nu_i} \le T, a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}} c_r(\nu_i, \nu_{i+1}, t_{\nu_i}) f(\nu_i, \nu_{i+1}, t_{\nu_i}) + \sum_{\nu_i \in V} \sum_{t_{\nu_i} \le T, a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}} c_w(\nu_i, t_{\nu_i}) f(\nu_i, t_{\nu_i}), \tag{22}$$

subject to:

$$\sum_{(s, \nu_i) \in A} \sum_{t_{\nu_i} \le T} f(s, \nu_i, t_{\nu_i}) = \sum_{(\nu_i, \rho) \in A} \sum_{\{t_{\nu_i} : t_{\nu_i} + b(\nu_i, \rho, t_{\nu_i}) \le T\}} f(\nu_i, \rho, t_{\nu_i}),$$

$$\forall \nu_i \in V, t_{\nu_i} \in [0, T], \tag{23}$$

$$\sum_{(\nu_i, \nu_{i+1}) \in A, t'_{\nu_i} + b(\nu_i, \nu_{i+1}, t'_{\nu_i}) = t_{\nu_i}} f(\nu_i, \nu_{i+1}, t'_{\nu_i}) + f(\nu_{i+1}, t_{\nu_{i+1}}) = \sum_{(\nu_{i+1}, \nu_i) \in A} f(\nu_{i+1}, \nu_i, t_{\nu_{i+1}}),$$

$$\forall \nu_i, \nu_{i+1} \in V \setminus \{s, \rho\}, t_{\nu_i} \in [0, T], a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}, i = 1, 2, \ldots, n, \tag{24}$$

$$0 \le f(\nu_i, \nu_{i+1}, t_{\nu_{i+1}}) \le l(\nu_i, \nu_{i+1}, t_{\nu_{i+1}}),$$

$$\forall (\nu_i, \nu_{i+1}) \in A, t_{\nu_i} \in [0, T], a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}, i = 1, 2, \ldots, n, \tag{25}$$

$$0 \le f(\nu_i, t_{\nu_i}) \le l(\nu_i, t_{\nu_i}) \forall \nu_i \in V, t_{\nu_i} \in [0, T], a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}, i = 1, 2, \ldots, n. \tag{26}$$

Equation (22) is the objective function that minimizes the total cost. Equation (23) is the flow value departure from the source s and arrival to the sink ρ with time $t_{\nu_i} \le T$. Equation (24) is flow conservation. Equations (25) and (26) are the capacity constraints on arcs and vertices, respectively. All constraints are satisfies the time-varying and time-windows conditions.

5. An Instance without a Waiting Time

Consider $G = (V, A, l, b, c_r, c_w, [a_{\nu_i}, b_{\nu_i}], s, \rho)$ be a network of a time-varying and time-windows with nonzero transit times, arbitrary costs, and no negative cycles. We will develop an algorithm to find a shortest dynamic f-augmenting path from s to ρ with time at most $t_{\nu_i} \le T$, where no waiting time $(w(\nu_i) = 0), \forall \nu_i \in V \setminus \{s, \rho\}, i = 1, 2, \ldots, n$ is permitted at any vertex. We will develop a procedure, called a shortest dynamic f-augmenting path of time-varying and time-windows searching process for the zero waiting time, which contains two different searching operations:

- the first is a forward searching,
- the second is a backward searching.

Both operations are designed by using the dynamic programming method, and the forward searching is to deal with positive transit times while backward searching will deal with negative transit times. The procedure will be solve the following sub-problem, where a transit times and a costs can be negative, and a negative cycle is defined as such a cycle that one can traverse at a time t_{ν_i} and return to the original place at the same time t_{ν_i}, where $t_{\nu_i} \in [0, T], a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}, i = 1, 2, \ldots, n$ with the total cost being negative.

Definition 5.1. Let $P(s, \nu_i)$ be a dynamic *f*-augmenting path with a time-varying and time-windows from s to $t_{\nu_i} \le T$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$. A section $P(\nu_i, \nu_j), 1 \le i \le j \le n$ is defined as a sub-path of $P(s, \nu_i)$, where all the transit times have the same sign. A section of $P(s, \nu_i)$ is said to be positive (or negative) if its transit times are all positive (or negative).

Remark. A dynamic *f*-augmenting path with a time-varying and time-windows consists of the several positive and negative sections alternatively. The number of these sections is said to be the alternating number of $P(s, \nu_i)$.

Definition 5.2. Let $d_z(\nu_i, t_{\nu_i})^k$ is the length of a shortest dynamic *f*-augmenting path with a time-varying and time-windows from the source vertex s to the vertex ν_i of time exactly $t_{\nu_i} \le T$, with the alternating number at most k.

Since there are no negative cycles in G, it is clear that a shortest dynamic faugmenting path P cannot contain more than n vertices and each vertex cannot be visited more than once at any time t_{ν_i}, $t_{\nu_i} \in [0, T]$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$; $i = 1, 2, \ldots, n$. Therefore, P cannot contain more than nT sections. In the other words, $d_z(\nu_i, t_{\nu_i})^k$ is the length of the shortest dynamic *f*-augmenting path with a time-varying and time-windows from s to ν_i of time exactly t_{ν_i}, when $k \ge nT$.

We now describe the procedure to solve the sub-problem, in which A^+ and A^- denote the set of all positive and negative arcs, respectively, where a positive (negative) arc means an arc with positive (negative) transit time.

In the following sections, we will present an algorithm of the MCFPTVTW; our algorithms satisfy the time-varying and time-windows constraint of the network problem without a waiting time.

6. An Algorithm of the MCFPTVTW

(i) The first algorithm:

Begin

Initialize: $d_z(s, 0)^o := 0$, $d_z(s, t_{\nu_i})^o := \infty$, $t_{\nu_i} \le T$,

$d_z(\nu_{i+1}, t_{\nu_{i+1}})^o := \infty$, $\forall \nu_{i+1} \in V \setminus \{s\}$; $t_{\nu_i} \le T$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$, $i = 0, 1, \ldots, n$;

Sort all values $u + b(\nu_i, \nu_{i+1}, u)$ for $u \in [1, T]$ and for all arcs $(\nu_i, \nu_{i+1}) \in A^+$;

Sort all values $u + b[\nu_i, \nu_{i+1}, u]$ for $u \in [0, T-1]$ and for all arcs $[\nu_i, \nu_{i+1}] \in A^-$; $i := 0$;

Do

$$i := i + 1;$$

For all $\nu_{i+1} \in V$, $t_{\nu_i} \in [0, T]$ **do** $d_z(\nu_{i+1}, t_{\nu_{i+1}})^i := d_z(\nu_{i+1}, t_{\nu_{i+1}})^{i-1}$

Case 1: i is an odd number:

For $1 \le t_{\nu_i} \le T$ **do**

For every $\nu_{i+1} \in V \setminus \{s\}$ **do** forward searching operation:

$$d_z(\nu_{i+1}, t_{\nu_{i+1}})^i := \min\{d_z(\nu_{i+1}, t_{\nu_{i+1}})^i, \min_{\{\nu_i : (\nu_i, \nu_{i+1}) \in A^+\}} \min_{\{u : u + b(\nu_i, \nu_{i+1}, u) = t_{\nu_i} \wedge l(\nu_i, \nu_{i+1}, u) > 0\}} \{d_z(\nu_i, u)^i + c_r(\nu_i, \nu_{i+1}, u)\}\};$$

Case 2: i is an even number:

For $t_{\nu_i} \in [T-1, 0]$ **do**

For every $\nu_{i+1} \in V \setminus \{\rho\}$ **do** backward searching operation:

$$d_z(\nu_{i+1}, t_{\nu_{i+1}})^i := \min\{d_z(\nu_{i+1}, t_{\nu_{i+1}})^i, \min_{\{\nu_i:(\nu_i, \nu_{i+1})\in A^-\}} \min_{\{u:u+b[\nu_i, \nu_{i+1}, u]=t_{\nu_i} \wedge l[\nu_i, \nu_{i+1}, u]>0\}} \{d_z(\nu_i, u)^i + c_r[\nu_i, \nu_{i+1}, u]\}\};$$

While there exists at least one $d_z(\nu_{i+1}, t_{\nu_{i+1}})^i \neq d_z(\nu_{i+1}, t_{\nu_{i+1}})^{i-1}$;

Let $d_z^*(\rho) = \min_{0\le t_{\nu_i}\le T} d_z(\rho, t_{\nu_i})^i$;

End

Theorem 6.1. *$d_z^*(\rho)$ is the length of a shortest dynamic f-augmenting path with a time-varying and time-windows from s to ρ at most T, $t_{\nu_i} \le T$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$ when the above algorithm with no waiting time is terminated.*

Proof. For each $\nu_i \in V$, $t_{\nu_i} \le T$, $a_{\nu_i} \le t_{\nu_i} \le b_{\nu_i}$, then $d_z(\nu_{i+1}, t_{\nu_{i+1}})^i$ obtained by the above algorithm which is the length of the shortest dynamic f-augmenting path from s to ν_{i+1} of a time-varying and time-windows exactly $t_{\nu_{i+1}}$ with the alternating number at most $i+1$. Furthermore, any shortest path must contain a positive section as it's the first section since there are no negative arcs $(s, \nu_{i+1}) \in A^-$ in the original network G or any residual networks. So, we only need to consider the paths whose first sections are positive.

The proof is carried out by double inductions on i, t_{ν_i}. Consider $i = 1$. Use the second induction on time $t_{\nu_i} \leq T$, $a_{\nu_i} \leq t_{\nu_i} \leq b_{\nu_i}$. When $t_{\nu_i} = 0$, since $i = 1$ is an odd number, no positive dynamic path $P(s, \nu_{i+1})$ of the time exactly t_{ν_i} exists in G except when $\nu_{i+1} = s$. We know that the length of the shortest dynamic path $P(s, s)$ of time 0 is 0. In the initialization of the above algorithm, we have $d_z(\nu_{i+1}, 0)^0 = \infty$, $\nu_{i+1} \in V \setminus \{s\}$ and $d_z(s, 0)^1 = 0$; hence the claim holds.

Assume that $t_{\nu_i} \succ 0$ and, for all values $t'_{\nu_i} < t_{\nu_i}$, $d_z(\nu_{i+1}, t'_{\nu_{i+1}})^1$ is the length of a shortest dynamic path $P(s, \nu_{i+1})$ of time exactly $t'_{\nu_{i+1}}$ with the alternating number at most 1 for all vertices ν_{i+1}.

Consider a vertex ν_{i+1}. First we prove that there exists a path of time exactly $t_{\nu_i} \leq T$ with the alternating number 1 and with length $d_z(\nu_{i+1}, t_{\nu_{i+1}})^1 = \infty$, there is nothing to prove. So assume $d_z(\nu_{i+1}, t_{\nu_{i+1}})^1$ is finite. Then by the forward searching operation, $d_z(\nu_{i+1}, t_{\nu_{i+1}})^i$ comes from $d_z(\nu_i, u)^1 + c_r(\nu_i, \nu_{i+1}, u)$ for some ν_i such that $(\nu_i, \nu_{i+1}) \in A^+$ and some u such that $u + b(\nu_i, \nu_{i+1}, u) = t_{\nu_{i+1}}$. By the induction on $t_{\nu_{i+1}}$, we know that there is a path $P' = (s = \nu_1, \ldots, \nu_{n-1} = \nu_i)$ of time exactly u with the alternating number at most 1 and with the length $d_z(\nu_i, u)^1$. Then, we extend the path with the vertex ν_{i+1}, obtaining a path $P(s, \nu_{i+1})$. The time of $P(s, \nu_{i+1})$ is exactly $a_{\nu_{i+1}} \leq t_{\nu_{i+1}} \leq b_{\nu_{i+1}}$ and the length is $d_z(\nu_i, u)^1 + c_r(\nu_i, \nu_{i+1}, u) = d_z(\nu_{i+1}, t_{\nu_{i+1}})^1$. This is what we want to obtain.

We now proof that $d_z(\nu_{i+1}, t_{\nu_{i+1}})^1$ is the length of the shortest path from s to ν_{i+1} of time exactly $t_{\nu_{i+1}} \leq T$. Let $P = (s = \nu_1, \ldots, \nu_n = \nu_{i+1})$ be the shortest path of the time-varying and time-windows exactly $t_{\nu_{i+1}}$ with the alternating number at most 1. If P is an empty path, then $\nu_{i+1} = s$ and the time of P is zero. Then the value $d_z(s, 0)^1 = 0$ is correct.

Let the path is not empty. Let ν_i be the predecessor of ν_{i+1} on this path. Let u be the time of the sub-path $P(s, \nu_i)$, and let $\psi(\nu_i)$ be the length of $P(s, \nu_i)$. By definition, $t_{\nu_{i+1}} = u + b(\nu_i, \nu_{i+1}, u)$. Since $u \prec t_{\nu_i}$, by the induction on t_{ν_i}, $\psi(\nu_i) \geq d_z(\nu_i, u)^1$. By the calculation formula, P is a path of shortest possible length and of the time exactly $t_{\nu_i} \leq T$, since there is exist a path of time exactly t_{ν_i} that achieves the length $d_z(\nu_{i+1}, t_{\nu_{i+1}})^1$, as we showed above. This completes the proof for time t_{ν_i} on $i = 1$.

Assume that for $i \prec k$, the claim is true. Now consider $i = k$. We discuss two cases in which i are an odd number and an even number, respectively. Suppose i is an odd number. Consider $t_{\nu_i} = 0$. Since no negative cycles in G, the length of the shortest dynamic path $P(s, s)$ of time exactly 0 with the alternating number at most i is 0. For the vertex $\nu_{i+1} \neq s$, since no flow can depart from ν_{i+1} at time 0, no residual network can contain a negative arcs which allow a flow to reach a vertex ν_{i+1} from any other vertex ν_i at time 0. This means that, there is no a dynamic path $P(s, \nu_{i+1})$ of time exactly 0. On the other hand, in the above algorithm, we let $d_z(\nu_{i+1}, 0)^i = d_z(\nu_{i+1}, 0)^{i-1}$ at first. By the induction on i we know $d_z(\nu_{i+1}, 0)^{i-1} = \infty$ and by the formula of the forward searching operation, $d_z(\nu_{i+1}, 0)^i$ are unchanged. Thus $d_z(\nu_{i+1}, 0)^i = +\infty$.

Assume the claim holds for $t'_{\nu_i} \prec t_{\nu_i}$. Now, consider the case at the time $t_{\nu_i} \leq T$. One can see that there exists a path of time exactly t_{ν_i} with the alternating number at most i and with the length $d_z(\nu_{i+1}, t_{\nu_{i+1}})^i$. We now prove that $d_z(\nu_{i+1}, t_{\nu_{i+1}})^i$ is the length of the shortest path from s to ν_{i+1} of the time exactly $t_{\nu_{i+1}}$. Let $P = (s = \nu_1, \ldots, \nu_n = \nu_{i+1})$ be a shortest path of the time exactly $t_{\nu_{i+1}}$ with the alternating number at most i. If P is an empty path, then $\nu_{i+1} = s$ and the time of P is 0. The value $d_z(s, 0)^i = 0$ is correct.

Assume that P is not empty. Let ν_i be the predecessor of ν_{i+1} on this path. Let u be the time of the sub-path $P(s, \nu_i)$, and let $\psi(\nu_i)$ be the length of $P(s, \nu_i)$. By definition, $t_{\nu_{i+1}} = u + b(\nu_i, \nu_{i+1}, u)$. By the induction, since $u \prec t_{\nu_i}$, $\psi(\nu_i) \geq d_z(\nu_i, u)^i$. By the definition, $P(s, \nu_{i+1})$ is a path of the shortest possible length and of the time exactly $t_{\nu_{i+1}} \leq T$, and since there exists a path of time exactly $t_{\nu_{i+1}}$ that achieves the length $d_z(\nu_{i+1}, t_{\nu_{i+1}})^i$. In the following, $\nu \in V$ represents of given a flow value to be sent from the source s to the sink ρ, and let f_j the maximal flow value which can be send along the j-th f-augmenting path with the time-varying and time windows $P_j(s, \rho)$.

- The first algorithm can be implemented in $O(nmT^2)$ time.

(ii) The second algorithm:

Begin

$$\overline{\nu} := 0;$$

For $j = 1, 2, \ldots, n$ **do**

Call the first algorithm;

If $d_z^*(\rho) \prec \infty$ then call the algorithm UP-NET; (there is an f-augmenting path $P_j(s, \rho)$ with the time-varying and time-windows of the flow value $f_j = Cap(P_j(s, \rho))$; so update the network)

Else stop; (no feasible solution to send all ν units of the flow from s to ρ within the time $t_{\nu_j} \le T$, $a_{\nu_j} \le t_{\nu_j} \le b_{\nu_j}$).

$$\overline{\nu} := \overline{\nu} + f_j;$$

If $\overline{\nu} \ge \nu$ then stop;

End

Theorem 6.2. *The second algorithm solves optimally the minimum cost flow problem with the time-varying and time-windows with no waiting times* $(w(\nu_i) = 0)$ *at each vertex* $\nu_i \in V \setminus \{s, \rho\}$.

Proof. It will suffice for us to prove that neither the original network G nor each the dynamic residual network contains any negative cycle. This will guarantee that the first algorithm can be applied correctly. For convenience, we denote the original network G by G_0 and the residual network generated after the j-th iteration by G_j. Suppose that G_j contains a negative cycles with the minimum subscript, and the cycle $C = (\nu_0, \nu_1', \ldots, \nu_l', \nu_n, \nu_{n-1}, \ldots, \nu_1, \nu_0)$ is one among these a negative cycles with the minimal length. We represent the shortest f-augmenting path $P_j(s, \rho) = (s, \ldots, \nu_0, \nu_1, \ldots, \nu_n, \ldots, \rho)$ found in the j-th iteration. The path $P' = (\nu_n, \nu_{n-1}, \ldots, \nu_o)$ is generated after the j-th iteration. The path $P(\nu_0, \nu_i)$ exists before the j-th iteration. Note that if we ignore the direction, P and P' become the same, which we denote as $\overline{P}$. We claim that $P_j(s, \rho)$ can only have one common sub-path $\overline{P}$ with C.

We prove the claim as follows; If $P_j(s, \rho)$ has at least two common sub-paths with C, then there must exist a cycle $C' = (\nu_i, \ldots, \nu_p, \ldots, \nu_k, \ldots, \nu_q, \ldots, \nu_i)$, since C' exists in G_{j-1}, it must have a positive length,

i.e., $\psi(C')=\psi(\nu_i, \ldots, \nu_p, \ldots, \nu_k) + \psi(\nu_k, \ldots, \nu_q, \ldots, \nu_i) \geq 0$. If $\psi(C') \succ 0$, we have $-\psi(\nu_i, \ldots, \nu_p, \ldots, \nu_k) \prec \psi(\nu_k, \ldots, \nu_q, \ldots, \nu_i)$. Then, a replacing $P(\nu_k, \ldots, \nu_q, \ldots, \nu_i)$ by $P(\nu_k, \ldots, \nu_p, \ldots, \nu_i)$ in C will be create a new cycle, and the length of this new cycle will be less than $\psi(C)$. However, we have assumed that C is the minimal one among all negative cycles. Thus we must have $\psi(C') = 0$, i.e., $-\psi(\nu_i, \ldots, \nu_p, \ldots, \nu_k) = \psi(\nu_k, \ldots, \nu_q, \ldots, \nu_i)$. Then, we can use $P(\nu_k, \ldots, \nu_p, \ldots, \nu_i)$ to replace $P(\nu_k, \ldots, \nu_q, \ldots, \nu_i)$ in C to eliminate cycle C' and combine two the common sub-paths into one. Following this operation, we can eliminate all other cycles and make $P_j(s, \rho)$ have only one common sub-path with C.

Now, we consider $C = (\nu_0, \nu'_1, \ldots, \nu'_l, \nu_n, \nu_{n-1}, \nu_1, \nu_0)$, $\psi(C) \succ 0$, we have $-\psi(\nu_n, \nu_{n-1}, \ldots, \nu_0) > \psi(\nu_0, \nu'_1, \ldots, \nu_n)$. On the other hand, we know $\psi(\nu_0, \nu_1, \ldots, \nu_n) = -\psi(\nu_n, \nu_{n-1}, \ldots, \nu_0)$, thus $\psi(\nu_0, \nu_1, \ldots, \nu_n) > \psi(\nu_0, \nu'_1, \ldots, \nu_n)$. Noting that $P_j(s, \rho)$ is the shortest path from s to ρ in G_{j-1} and $P(\nu_0, \nu_1, \ldots, \nu_n)$, $P(\nu_0, \nu'_1, \ldots, \nu_n)$ exist in G_{j-1}, we can use $P(\nu_0, \nu'_1, \ldots, \nu_n)$ to replace $P(\nu_0, \nu_1, \ldots, \nu_n)$ in $P_j(s, \rho)$ to obtain another path, $P'(s, \rho)$, with the property that $\psi(P') < \psi(P)$. This is a contradiction. Therefore, G_j can not contain any negative cycles.

- The second algorithm can be implemented in $O(nmVT^2)$ time.

7. Conclusion

We consider the minimum cost flow problem on a time-varying and time-windows (MCFPTVTW) which is a new version of the minimum cost flow problem (MCFP). The objective is to find an optimal schedule to send a flow from the source vertex to the sink vertex with the time-varying and time-windows subject to the constraint that the flow must arrive at the sink vertex before a deadline T. We propose a mathematical model of the MCFPTVTW. We give an instance without a waiting time. Finally, an algorithm of the MCFPTVTW is presented.

Acknowledgements

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

References

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

[2] R. K. Ahuja, A. V. Goldberg, J. B. Orlin and R. E. Tarjan, Finding minimum-cost flows by double scaling, Mathematical Programming 53(1-3) (1992), 243-266.

DOI: https://doi.org/10.1007/BF01585705

[3] R. G. Busacker and P. J. Gowen, A Procedure for Determining a Family of Minimum Cost Network Flow Patterns, Technical Report 15, Operation Research, Johns Hopkins University, 1961.

[4] X. Cai, T. Kloks and C. K. Wong, Time-varying shortest path problems with constraints, Networks 29(3) (1997), 141-150.

DOI: https://doi.org/10.1002/(SICI)1097-0037(199705)29:3<141::AID-NET2>3.0.CO;2-H

[5] L. Ciupala and E. Ciurea, About perflow algorithms for the minimum flow problem, WSEAS Transactions on Computer Research 3(1) (2008), 35-42.

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

[7] J. Edmonds and R. M. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, Journal of the ACM 19(2) (1972), 248-264.

DOI: https://doi.org/10.1145/321694.321699

[8] N. A. El-Sherbeny, Resolution of a Vehicle Routing Problem with a Multi-Objective Simulated Annealing Method, Ph.D. Dissertation, Faculty of Science, Mons University, Mons, Belgium, 2002.

[9] N. A. El-Sherbeny, Algorithm of fuzzy maximum flow problem with fuzzy time-windows in hyper network, International Journal of Pure and Applied Mathematics 116(4) (2017), 863-874.

DOI: http://dx.doi.org/10.12732/ijpam.v116i4.6

[10] N. El-Sherbeny and D. Tuyttens, Optimization Multicriteria of Routing Problem, Troisieme Journée de Travail sur la Programming Mathématique Multi-Objective, Faculté Polytechnique de Mons, Mons, Belgique, 2001.

[11] N. A. El-Sherbeny, Imprecision and flexible constraints in fuzzy vehicle routing problem, American Journal of Mathematical and Management Sciences 31(1-2) (2011), 55-71.

DO: https://doi.org/10.1080/01966324.2011.10737800

[12] L. R. Ford and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, NJ, 1962.

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

DOI: http://dx.doi.org/10.4153/CJM-1956-045-5

[14] A. V. Goldberg and R. E. Tarjan, Finding minimum-cost circulations by successive approximation, Mathematics of Operation Research 15(3) (1990), 430-466.

DOI: https://doi.org/10.1287/moor.15.3.430

[15] D. Lozovanu and S. Pickl, A special dynamic programming technique for multiobjective discrete control and for dynamic games on graph-based methods, CTWO4 Workshop on Graphs and Combinatorial Optimization, Milano, (2004), 184-188.

[16] 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, Springer-Verlag, Berlin, Germany, 535 (2004), 133-152.

DOI: https://doi.org/10.1007/978-3-642-17144-4_5

■