International Journal of Sciences: Basic and Applied Research (IJSBAR)

ISSN 2307-4531
(Print & Online)

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

Difference of Convex Functions Optimization Methods: Algorithm of Minimum Maximal Network Flow Problem with 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, Turabah, KSA

Email: nasserelsherbeny@yahoo.com

Abstract

In this paper, we consider the minimum maximal network flow problem, i.e., minimizing the flow value, minimizing the total time among maximal flow with time-windows, which is a combinatorial optimization and an NP-hard problem. After a mathematical modeling problem, we introduce some formulations of the problem and one of them is a minimization of a concave function over a convex set. The problem can also be cast into a difference of convex functions programming (nonconvex optimization). We propose in this work a new algorithm for solving the Minimum Maximal Network Flow Problem with Time-Windows (MMNFPTW).

***Keywords*:** Optimization network; Minimum flow problem; Difference of convex functions optimization; Time-windows.

1. Introduction

In recent years there has been a very active research in difference of convex functions programming (nonconvex optimization), because most of real life optimization problems are nonconvex.

---

* Corresponding author.

The field of network optimization flows has a rich and long history, a difference of convex functions programming and a difference of convex functions algorithms introduced by Pham Donh Tao in 1985. Such early work established the foundation of the key ideas of network optimization flow theory. The key task of this filed is to answer such questions as, which way to use a network is most cost effective? Maximum flow problem and minimum cost flow problem are two typical problems of them. However, from the point of view of practical cases, we have another kind of problems which are inherently different form the typical ones. For instance, Figure 1 and 2 portrays a network with edge flow capacity one unit on all edges, each edge has a transit time $t_{ij} \in \Re^+, i \neq j, i, j = 1,2,...,n.$ For each vertex $i \in V,$ a time-windows $[a_i, b_i]$ within which the vertex may be served and $a_i \leq t_i \leq b_i,$ $t_i \in T$ is a nonnegative service and leaving time of the vertex. A source vertex $s,$ with time windows $[a_s, b_s]$, a sink vertex τ with time-windows $[a_\tau, b_\tau]$ and t_s is a departure time of the source vertex see [1,2,3,4,5,6,7,15].

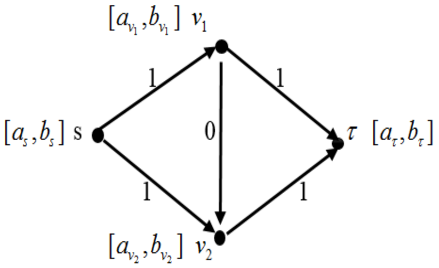

Figure 1

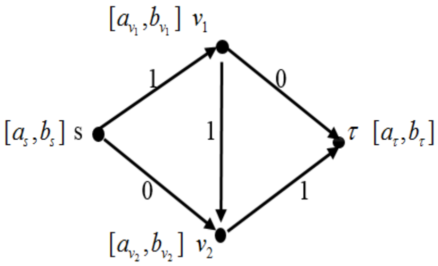

Figure 2

Minimum Maximal Network Flow Problem with Time-Windows (MMNFPTW)

- The figure 1 illustrates the maximum flow of the network, that is, the flow on all edges is one except the edge $x_3,$ whose flow is zero. On the other hand, if the flow on x_3 is fixed at one and we cannot reduce it by some reasons such as emergency, then the network cannot be exploited at the most economical situation. In this case, we can send two unit of flow from a source vertex s to a sink vertex τ which satisfy a time-windows constraint.
- In the figure 2, the flow on x_3 is fixed at one, the possible flow value we can send between s and τ is one unit.

The flow value, we can send between s and τ reduces from two (in figure 1) to one (in figure 2) due to fact that the flow value of x_3 is undirected. It means that the maximum flow value is not attainable if the users on the network are disobedient.

Form the point of view of modeling, the above two figures cases are essentially different though they bear some resemblance. Assuming that the flow is directly, the figure 1 aims at an optimal value flow. The figure 2 also searches for an optimal value of flow, without the directly of a network flow. The standard network flow with the directly has been well studied for several decades. Without the directly, many problems in network flow, the maximum flow problem, become more difficulty. Compared to the standard network flow theory is a new filed, hence is still in its infancy.

The natural question in this new field is: given a network $N,$ how to calculate the attainable maximum value flow of N when the flow is undirected. To answer the question, in this paper we consider minimum maximal network flow problem with time-windows which finds out the minimum value and minimum total time satisfy a time-windows constraint among the maximal flows in the network N from a source vertex s to a sink vertex $\tau.$

Iri [12] gave the definition of undirected flow (u-flow) and presented fundamental problems related u-flow. Although the concept of u-flow is quite different from maximal flow and their relationship is not known yet so much, the optimal value of minimum maximal u-flow of a network N is equal to the optimal value of minimum maximal flow under some assumption. In Iri [8] profound essay, several fundamental theorems and new research topics are described, but no algorithms for the corresponding problems are proposed. To the author's knowledge, no algorithms for the minimum maximal flow were known until Shi-Yamamoto [12]. As pointed out in [14], Shi-Yamamoto's algorithm is not efficient enough. After that, some algorithms for solving the problem were proposed in such as Shigeno-Yamamoto [13] and others. Since the theory dealing with the network flow problems without assuming the amenability of flows is still in its infancy, in this paper we focus on the development of algorithm for Minimum Maximal Network Flow Problem with Time-Windows (MMNFPTW) in virtue of difference convex functions optimization.

The reminder of this work is organized as follows. In Section 2, we give the mathematical models of the problem and its equivalent formulations. In Section 3, we then outline the properties of the difference convex programming and a difference convex algorithm. We describe the framework of the difference convex algorithm with time-windows. In Section 4, we give a new algorithm of a difference convex Minimum Maximal Network Flow Problem with Time-Windows (MMNFPTW). Finally, the conclusion is given in Section 5.

2. Mathematical Models and Equivalent Formulations

2.1 Basic Concepts and Definitions

Consider a directed network $N=(V,E),$ where V is a set of $m+2$ vertices, E is a set of n edges with a non-negative transit time $t_{ij},\ i\neq j, i, j\in V.$ For each vertex $i\in V,$ a time windows $[a_i,b_i]$ within which the vertex may be served and $a_i\leq t_i\leq b_i,\ t_i\in T$ is a non-negative service and leaving time of the vertex $i.$ A single source vertex $s,$ a single sink vertex τ with time windows $[a_s,b_s],[a_\tau,b_\tau]$ respectively, and c is the vector of the edge capacity. Let X denote the set of feasible flow,

$$X = \{x \mid x \in \Re^n, Ax = 0, 0 \le x \le c\} \quad (1)$$

where the matrix A stands for a vertex edge incident relationship in the network. Obviously, X is a compact convex set.

Definition 2.1.1 A vector $z \in X$ is said to be maximal flow if there does not exist $x \in X$ such that $x \ge z$ and $x \ne z.$

Let f be the flow value function, f is assumed to be linear on $X.$ For instant, it usually fined by

$$f(x) = \sum_{i \in \delta^+(s)} x_i - \sum_{i \in \delta^-(s)} x_i \quad (2)$$

where $\delta^+(s)$ and $\delta^-(s)$ are the sets of edges which leaves and enters the source vertex $s,$ respectively. Then f is a linear function. Let $d \in \Re^n$ where, $d_i = 1,$ if $i \in \delta^+(s)$, $d_i = -1,$ if $i \in \delta^-(s)$ and $d_i = 0$, if otherwise.

In this work, let $\Re^k$ denotes the set of *k*-dimensional real column vectors, $\Re^k_* = \{x \mid x \in \Re^k; x \ge 0\}$ and $\Re^k_{**} = \{x \mid x \in \Re^k; x \succ 0\}.$ Let $\Re_k$ denotes the set of *k*-dimensional real row vectors, and $\Re^*_k = \{x \mid x \in \Re_k; x \ge 0\},$ $\Re^{**}_k = \{x \mid x \in \Re_k; x \succ 0\}.$ We denote e to both a row vector and a column vector of ones, and e_i to denote the $i^{\underline{th}}$ with row vector of an appropriate dimension. For a set $S,$ $V(S)$ is the set of extreme points of $S.$ Let X_M denote the set of all maximal flows,

$$X_M = \{z \in X \mid \text{there does not exist } x \in X \text{ such that } x \ge z, \text{ and } x \ne z\} \quad (3)$$

We consider the problem is given by

$$\min\{f(x) \mid x \in X_M\} \quad (4)$$

Let X_E denote the efficient set of the vector optimization problem, then the problem (4) is equivalent to the problem

$$\min\{f(x) \mid x \in X_E\} \quad (5)$$

2.2 Primal Formulation Model

We define the function r as;

$$r(x) = \max\{e(y - x) \mid y \geq x, y \in X\} \tag{6}$$

Clearly, $r(x) \geq 0, \forall x \in X$. It is easy to see that r is a concave function on X. In fact, $\forall \alpha \in (0,1)$ and $x_1, x_2 \in X$ we have,

$$\begin{aligned}
&\alpha r(x_1) + (1-\alpha) r(x_2) = \\
&= \alpha \max\{e(y - x_1) \mid y \geq x_1, y \in X\} + (1-\alpha)\max\{e(y - x_2) \mid y \geq x_2, y \in X\} \\
&= \alpha e(y_{r(x_1)} - x_1) + (1-\alpha) e(y_{r(x_2)} - x_2) \\
&= e(\alpha y_1 + (1-\alpha) y_2 - (\alpha x_1 + (1-\alpha) x_2)) \\
&\leq \max\{e(y - (\alpha x_1 + (1-\alpha) x_2)) \mid y \geq \alpha x_1 + (1-\alpha) x_2, y \in X\} \\
&= r(\alpha x_1 + (1-\alpha) x_2),
\end{aligned} \tag{7}$$

where $y_{r(x)} \in \arg\max\{e(y - x) \mid x, y \in X\}$. Moreover, $r(x)$ is piecewise linear on X. In fact, adding a slack z such that,

$$\begin{pmatrix} A & 0 & 0 \\ I & I & 0 \\ I & 0 & -I \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ c \\ x \end{pmatrix}, \; z \in \Re_*^{2n} \Leftrightarrow Ay = 0, x \leq y \leq c \tag{8}$$

Then for a given $x \in \Re_*^n, r(x)$ is a solution of the following linear programming:

$$\max ey - ex$$

$$\textit{subject to} \quad \begin{pmatrix} A & 0 & 0 \\ E & E & 0 \\ E & 0 & -E \end{pmatrix} \begin{pmatrix} y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ c \\ x \end{pmatrix}, y \geq 0, z \geq 0 \tag{9}$$

where E is an $n \times n$ matrix. As $r(x)$ is a solution of a linear maximization, we assume that

$$r(x) = c_B B^{-1} \begin{pmatrix} 0 \\ c \\ x \end{pmatrix} - ex \tag{10}$$

where c_B a corresponding coefficient is vector of objective function, and B is a basic matrix of problem (9).

Lemma 2.2.1 If the capacity c is integral and satisfy the time-windows constraint, then so is $r(x)$ for any integer x.

Proof: Obviously, because of the concave function, we assume that

$$r(x) = \min\{l_i(x) \mid i \in I \subseteq V, a_i \le t_i \le b_i, t_i \in T\} \tag{11}$$

where $t_i + t_{ij} \le t_j, \forall i, j \in V, t_i, t_{ij} \in T$ and $l_i(x) = \langle l_i, x\rangle + \gamma_i$ are linear function on $\Re^n$. It is easy to see that $x \in X, r(x) = 0$ if and only if $x \in X_E$. Hence, problem (4) can be rewritten equivalently as

$$\min\{f(x) \mid x \in X, r(x) \le 0\} \tag{12}$$

Consider the following penalized problem for a fixed number u.

$$\min\{f(x) + ur(x) \mid x \in X\} \tag{13}$$

Then, we proof the following lemma

Lemma 2.2.2 Let $u_* = \max\{f(x) \mid x \in X\} - \min\{f(x) \mid x \in X\}$, then there exists a finite number $u_* \ge 0$, such that for every $u \succ u_*$, the problem (12) is equivalent to the problem (13).

Proof: For any $u \succ u_*$ and $r(x) \ne 0$, we must have $r(x) \ge 1$ and

$$\begin{aligned} \min\{f(x) + ur(x) \mid x \in X\} &\ge \min\{f(x) + u_* r(x) \mid x \in X\} \\ &\ge \min\{f(x) \mid x \in X\} + \max\{f(x) \mid x \in X\} - \min\{f(x) \mid x \in X\} \\ &\ge \min\{f(x) \mid x \in X, r(x) = 0\} \end{aligned} \tag{14}$$

and when $r(x_0) = 0$ then we see, $f(x_0) \ge \min\{f(x) \mid x \in X, r(x) = 0\}$. Hence,

$$\min\{f(x) + ur(x) \mid x \in X\} \ge \min\{f(x) \mid x \in X, r(x) = 0\} \tag{15}$$

On the other hand, a feasible solution of $\min\{f(x) \mid x \in X, r(x) = 0\}$ is also a feasible solution of (13). We have

$$\min\{f(x) + ur(x) \mid x \in X\} \le \min\{f(x) \mid x \in X, r(x) = 0\} \tag{16}$$

It implies that

$$\min\{f(x) + ur(x) \mid x \in X\} = \min\{f(x) \mid x \in X, r(x) = 0\} \tag{17}$$

We note that, $r(.) \geq 0,$ then any $u \succ u_* \geq 0$ we have

$$\min\{f(x)+ur(x) \mid x \in X\} = \min\{f(x) \mid x \in X, r(x)=0\}$$

$$= \min\{f(x) \mid x \in X, r(x) \leq 0\} \tag{18}$$

Then, (12) is equivalent to (13).

New, we denote by δ_X the indicator of X and $g(x) = f(x) + \delta_X(x),$ also,

$$h(x) = \begin{cases} -u_* r(x); x \in X \\ \infty; x \notin X \end{cases} \tag{19}$$

Then $g(x)$ and $h(x)$ are convex and problem (4) is rewritten as

$$\min\{f(x) + \delta_X(x) - h(x)\} = \min\{g(x) - h(x)\} \tag{20}$$

This is a difference of convex functions programming. Hereafter, we use the formation for local search in a difference of convex functions algorithms.

2.3 Dual Formulation Model

In Philip [11], it follows that there exists a simplex $\Lambda \subseteq \Re_n$ such that a vector x is maximal flow if and only if there exists $\lambda \in \Lambda$ such that

$$\lambda x \geq \lambda y, \forall y \in X \tag{21}$$

Thus, the minimum maximal flow problem with time-windows to be considered can also be formulated as:

$$\min f(x)$$

$$\textit{subject to } -\lambda(y-x) \geq 0, \forall y \in X, \ \lambda \in \Lambda, x \in X \tag{22}$$

$$t_i + t_{ij} \leq t_j, \ a_i \leq t_i \leq b_i, \ t_i, t_{ij} \in T, i \neq j, \ \forall i, j \in V$$

This is a special case of mathematical programming with variation inequality and time-windows constraint.

We denote that, $g(x,\lambda) = \frac{1}{2}\|x\|^2 + \frac{1}{2}\|\lambda\|^2 + \max_{v \in X}\{vx + v\lambda - \frac{1}{2}\|v\|^2\}$ (23)

and $$h(x,\lambda)=\frac{1}{2}\|x+\lambda\|^2+\frac{1}{2}\|x\|^2 \qquad (24)$$

Then, we proof the following lemma

Lemma 2.3.1 The constraints in (22) can be cast into the form

$$g(x,\lambda)-h(x,\lambda)=0, \lambda\in\Lambda, x\in X \qquad (25)$$

Proof: We note that, $g(x,\lambda)-h(x,\lambda)=\max_{v\in X}\{\lambda(v-x)-\frac{1}{2}\|v-x\|^2\}$ (26)

Since X is a convex set. Suppose that (21) holds for some $x\in X$ and some $\lambda\in\Lambda,$ we have that;

$$0\le\max_{v\in X}\{\lambda v-\lambda x-\frac{1}{2}\|v-x\|^2\}\le\max\{\lambda v-\lambda x \mid v\in X\}=0 \qquad (27)$$

which yields $g(x,\lambda)-h(x,\lambda)=0$

Suppose that $g(x,\lambda)-h(x,\lambda)=0$ for some $x\in X$ and $\lambda\in\Lambda.$ Then we have that,

$$\max_{v\in X}\{\lambda(v-x)-\frac{1}{2}\|v-x\|^2\}=0 \qquad (28)$$

which implies that $\lambda(v-x)\le 0$ for all $v\in X.$ In fact, if we have some $v_0\in X$ such that $\lambda(v_0-x)\succ 0,$ then we can take a point $\bar{v}$ on line segment $[v_0,x]$ satisfying $\|\bar{v}-x\|\prec\|\lambda\|\cos\theta,$ where θ is the acute angel between λ and $v_0-x.$ Since X is convex, then $\bar{v}\in X$ but $\lambda(\bar{v}-x)-\frac{1}{2}\|\bar{v}-x\|^2\succ 0.$ It contradicts (28).

Note that: The functions g and h are convex and differentiable.

From lemma 2.3.1, it follows that the problem can be formulated by the following a difference of convex functions of differentiable programming with time-windows constraint:

$$\min f(x)$$

subject to $g(x,\lambda)-h(x,\lambda)=0$, $\lambda\in\Lambda, x\in X$ (29)

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

By Shigeno-Takahashi-Yamamoto [11], we see that the Λ in (29) could be replaced by $\{\lambda \mid \lambda \in \Re_n^{**}, \lambda \geq e, \lambda e = n^2\}$. Then we take the above set as Λ to design the algorithms.

3. A Difference Convex Programming and a Difference Convex Algorithm

A difference convex programming and a difference convex algorithms introduced by Pham Dinh Tao in 1985 and extensively developed in other works. A difference convex algorithms was successfully applied to a lot of different and various nonconvex optimization problems to which it quite often gave a global solutions and proved to be more robust and more efficient than related standard methods, especially in the large scale setting.

In [10] a difference convex algorithm is a primal-dual approach for finding local optimum in a difference convex programming. More detailed results on a difference convex algorithms can be found in such as [9]. Some numerical experiments are reported that it finds a global minimizer often if one chose a good start point.

Consider the following general problem:

$$v_p = \inf\{g(x) - h(x) \mid x \in \Re^n\} \tag{30}$$

where $g(.), h(.): \Re^n \to \Re \cup \{-\infty, \infty\}$ are low semi-continuous convex functions on $\Re^n$. It is easy to see that problem (20) is a special case of (30) as shown in (20) under the conservation $+\infty$. We also suppose that $g(x) - h(x)$ is bounded below on $\Re^n$. The ε -subgradient of g at point x_0 are defined by:

$$\partial_\varepsilon g(x_0) = \{y \in \Re^n \mid g(x) \geq g(x_0) + \langle x - x_0, y\rangle - \varepsilon, \forall x \in X\} \tag{31}$$

and $\partial g(x_0) = \partial_0 g(x_0)$. The conjugate function of g is given by:

$$g^*(y) = \sup\{\langle x, y\rangle - g(x) \mid x \in \Re^n\} \tag{32}$$

From low semi-continuous of g and h, we see that $g = g^{**}$ and $h = h^{**}$ hold. Consider a dual problem of (30):

$$v_d = \inf\{h^*(y) - g^*(y) \mid y \in \Re^n\} \tag{33}$$

We have that $v_p = \inf\{g(x) - h(x) \mid x \in \Re^n\}$

$$= \inf\{g(x) - \sup\{\langle x, y\rangle - h^*(y) \mid y \in \Re^n\} \mid x \in \Re^n\}$$

$$= \inf\{g(x) + \inf\{h^*(y) - \langle x, y\rangle \mid y \in \Re^n\} \mid x \in \Re^n\}$$

$$= \inf\{h^*(y) + \inf\{g(x) - \langle x, y\rangle \mid x \in X\} \mid y \in \Re^n\} \qquad (34)$$

$$= \inf\{h^*(y) + \sup\{\langle x, y\rangle - g(x) \mid x \in X\} \mid y \in \Re^n\}$$

$$= \inf\{h^*(y) - g^*(y) \mid y \in \Re^n\} = v_d$$

For a pair $(x, y),$ Fenchel's inequality $g(x) + g^*(y) \geq \langle x, y\rangle$ holds for any proper convex function g and g^*. If $y \in \partial g(x)$ then $g(x) + g^*(y) = \langle x, y\rangle.$

Definition 3.1 A point x^* is said to be local minimal of $g - h$ if there exists a neighborhood N of x^* such that $(g-h)(x) \geq (g-h)(x^*), \forall x \in N$

Lemma 3.2 A point x^* is local minimal for $g - h,$ then $\partial h(x^*) \subseteq \partial g(x^*).$

Proof: Let $(g-h)(x) \geq (g-h)(x^*), \forall x \in N.$ Then $g(x) - g(x^*) \geq h(x) - h(x^*).$ Taking $z \in \partial h(x^*),$ we have $h(x) \geq h(x^*) + \langle x - x^*, z\rangle$ for all $x \in \Re^n$. Therefor, we see that $g(x) \geq g(x^*) + \langle x - x^*, z\rangle$ for $x \in N.$ We note that g is convex, then $g(x) \geq g(x^*) + \langle x - x^*, z\rangle$ holds for $x \in \Re^n.$

Lemma 3.3 If h is a piecewise linear convex function on $dom(h)$ and $\partial h(x^*) \subseteq \partial g(x^*),$ then x^* is local minimal for $g - h.$

Proof: It is enough to consider $x \in dom(g).$ Let h is piecewise linear convex. Then there exist a neighborhood $N(x^*)$ such that for any $x \in N(x^*)$ we can choose $z \in \partial h(x^*)$ such that $h(x) - h(x^*) = \langle x - x^*, z\rangle.$ For $\partial h(x^*) \subseteq \partial g(x^*)$ we have $g(x) \geq g(x^*) + \langle x - x^*, z\rangle$ holds for $x \in N(x^*).$ It implies that $g(x) - h(x) \geq g(x^*) - h(x^*)$ for $x \in N(x^*).$ Then x^* is local minimal for $g - h.$

- **A Difference Convex Algorithm with Time-Windows**

We describe a framework of the Difference Convex Algorithm with Time-Windows is the first algorithm.

0: pick up a point $x^0 \in dom(h),$ calculate $y^0 \in \partial h(x^0); k = 1;$

1: each point has satisfied a time-windows constraint, i.e.,

$t_i + t_{ij} \le t_j$, $a_i \le t_i \le b_i$, $t_i, t_{ij} \in T, i \ne j$, $\forall i, j \in V$;

2: calculate $x^k \in \arg\min\{g(x) - (h(x^{k-1}) + \langle x - x^{k-1}, y^{k-1}\rangle) \mid x \in \Re^n\}$;

calculate $y^k \in \arg\min\{h^*(y) - (g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle) \mid y \in \Re^n\}$;

3: If $\partial h(x^k) \cap \partial g(x^k) \ne \phi,$ stop; otherwise, $k = k+1$ go to step 1.

Lemma 3.4 Suppose that the points x^k and y^k are satisfied a time windows-constraint and generated in the above first algorithm, then $x^k \in \partial h^*(y^k)$ and $y^{k-1} \in \partial g(x^k).$

Proof: Assume that x^{k-1} and y^{k-1} are satisfied a time windows constraint and in hand.

We have $\min\{g(x) - (h(x^{k-1}) + \langle x - x^{k-1}, y^{k-1}\rangle) \mid x \in \Re^n\}$

$= \min\{g(x) - \langle x, y^{k-1}\rangle \mid x \in \Re^n\} - h(x^{k-1}) + \langle x^{k-1}, y^{k-1}\rangle$ and

$$\min\{h^*(y) - (g^*(y^{k-1}) + \langle x^k, y - y^{k-1}\rangle) \mid y \in \Re^n\}$$

$$= \min\{h^*(y) - \langle x^k, y\rangle) \mid y \in \Re^n\} - g^*(y^{k-1}) + \langle x^k, y^{k-1}\rangle. \tag{35}$$

Thus, from step 2 in the above first algorithm, $g(x) - \langle x, y^{k-1}\rangle \ge g(x^k) - \langle x^k, y^{k-1}\rangle$ for all $x,$ and $h^*(y) - \langle x^k, y\rangle \ge h^*(y^k) - \langle x^k, y^k\rangle$ for all $y.$ It yields $y^{k-1} \in \partial g(x^k)$ and $x^k \in \partial h^*(y^k).$

4. Methods and Algorithm

Now we go back to problem (20). In this section, we give an algorithms to solve the problem. A General Framework of branch-and-bound algorithm with time-windows can be stated follows.

- Algorithm General Framework:

0: initial setting and calculating;

1: branching operation with time-windows constraint, i.e.,

$t_i + t_{ij} \le t_j$, $a_i \le t_i \le b_i$, $t_i, t_{ij} \in T, i \ne j$, $\forall i, j \in V$;

2: local search for a smaller upper bound;

3: find a larger lower bound;

4: remove some regions, do to step 1.

We describe the step 1, 2 and 3 as the following explained:

- Describe step 1: Branching operation with time-windows constraint

A simplex based division is usually exploited in branch-and-bound method. At some step, a contemporary simplex S is divided into two smaller ones S_1 and S_2. Taking into account the convergence of the algorithms and a time-windows constraint, we need the division to be exhaustive, i.e., a nested sequence of simplexes $\{S_k\}, k = 1,2,...$ has the following properties:

1. $\text{int}(S_i) \cap \text{int}(S_j) = \phi$ if $i \neq j;\ i, j \in V$ has a time windows $[a_i, b_i],[a_j, b_j]$ where $a_i \leq t_i \leq b_i$, $a_j \leq t_j \leq b_j$ respectively and $t_i + t_{ij} \leq t_j$, $S_{k+1} \subseteq S_k$ for all k,
2. $\lim\limits_{k\to\infty} \cap_{k=1}^{\infty} S_k = x^0$ for some x^0.

At each step, we chose divide a simplex S_k into two smaller ones S_{2k} and S_{2k+1} by bisecting the longest edge of S_k. The sequence $\{S_k\}, k = 1,2,...$ in such process is exhaustive.

- Describe step 2: Local search for a smaller upper bound

There are many methods to do local search. Here we exploit the first algorithm in this step. Even the first algorithm is not going to find a global optimum theoretically, but in many numerical experiments, it finds a global optimum practically.

As shown in problem (20) can be rewritten as a difference convex programming $\min\{g - h\}$, then we can use the first algorithm to obtain a locally optimal solution. Then we assume that $o^i = X \cap S_i$ of

the first algorithm is a local optimal solution satisfy the time-windows constraint on $X \cap S_i$ by using the first algorithm.

- Describe step 3: Find a larger lower bound

Assume that $l_i(x)$ is an affine function such that $l_i(v_j) = h(v_j)$ for all vertices $v_j \in V(S_i)$ with time-windows $[a_{v_j}, b_{v_j}]$, $a_{v_j} \leq t_{v_j} \leq b_{v_j}$, $t_{v_j} \in T$ be a non-negative time. From the convex of $h(x)$, we have $l_i(x) \geq h(x)$ for all $x \in S_i$. Then,

$$L(X \cap S_i) = \min\{f(x) + \delta_{X \cap S_i}(x) - l_i(x) \mid x \in \Re^n\}$$

$$\leq \min\{f(x) + \delta_{X \cap S_i}(x) - h(x) \mid x \in \Re^n\} \qquad (36)$$

Moreover, if $V(S_i) = \{v_1, ..., v_{p_i}\}$ is in hand and satisfy the time-windows constraint, then it is easy to calculate $L(X \cap S_i)$ because

$$\min\{f(x) + \delta_{X \cap S_i}(x) - l_i(x) \mid x \in \Re^n\}$$

$$= \min\{d\sum_{j=1}^{p_i} \lambda_j v_j + u_* \sum_{j=1}^{p_i} \lambda_j r(v_j) \mid \sum_j \lambda_j = 1, \lambda_j \geq 0, A\sum_{j=1}^{p_i} \lambda_j v_j = b, 0 \leq \sum_{j=1}^{p_i} \lambda_j v_j \leq c\} \qquad (37)$$

Based on the above discussion, we give the following algorithm of the difference convex algorithm of the Minimal Maximum Network Flow Problem with Time-Windows (MMNFPTW).

- The Difference Convex Algorithm of the Minimal Maximum Network Flow Problem with Time-Windows

0: let ε and S_0 such that $X \subseteq S_0$. let $x^0 = 0, y^0 = (-1, ..., -1)$, $b^U = basiDCA(x)$, $b_L = \min\{f(x) \mid Ax = b, 0 \leq x \leq b\}, M = S_0$;

1: select $S_0 \in M$ such that $b_L = L(X \cap S_0)$ and dived S_0 into S_1 and S_2;

2: $o^i = X \cap S_i$ from the first algorithm for all $i = 1,2$ if $o^i \prec b^U$ then $o^i = b^U$;

3: if $L(X \cap S_i) \succ b_L$ then $b_L = L(X \cap S_i)$, if $b^U - b^L \prec \varepsilon$ then Stop;

4: $M = \{S \in M \mid L(X \cap S) \prec b^U\}$, if $M = \phi$ then Stop, otherwise, go to step 1.

The convergence of the above new algorithm of the Minimum Maximal Network Flow Problem with Time Windows (MMNFPTW) is from the exhaustive partition.

5. Conclusion

A branch-and-bound algorithm via a difference convex algorithm subroutine for solving problem (4) is proposed in this work. A part from the algorithm, we also discussed a dual formulation for problem (4) and investigated some properties of a general difference convex programming. Though we have not proposed an algorithm for problem (30), it can be solved by differential programming. Due to that problem (13) is a concave minimization over a convex set, we can solve it by many existing methods directly or indirectly. Among these methods, it might be interesting to compare the behavior of the different algorithms. Also, we introduce a new algorithm of the Minimum Maximal Network Flow Problem with Time-Windows (MMNFPTW).

Acknowledgment

The author would like to thank the anonymous referees for their comments and careful readings.

References

[1] N. El-Sherbeny. (2015). "Minimum Convex and Differentiable Cost Flow Problem with Time- Windows". International Journal of Sciences: Basic and Applied Research, Vol. (20), No. (1), pp. 139-150. Available: http://gssrr.org/index.php?journal=JournalOfBasicAndApplied

[2] N. El-Sherbeny. (2015). "The Dynamic Shortest Path Problems of Minimum Cost Length Time Windows and Time-Varying Costs". International Journal of Scientific and Innovative Mathematical Research, Vol. (3), No. (3), pp. 47-55. Available: www.arcjournal.org

[3] N. El-Sherbeny. (2014). "The Algorithm of the Time-Dependent Shortest Path Problem with Time-Windows". Applied Mathematics, Vol. (5), No. (17), pp. 2764-2770. Available: http://dx.doi.org/10.4236/am.2014.517264

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

[5] N. El-Sherbeny. And D. Tuyttens. "Optimization multicriteria of routing problem". Troisieme Journee de Travail sur la Programming Mathematique Multi-Objective, Faculte Polytechnique de Mons, Mons, Belgique, (2001).

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

[7] R. Horst and H. Tuy. (1993). Global optimization, Deterministic Approaches. (2nd edition). Springer-Verlag.

[8] M. Iri. "An essay in the theory of uncontrollable flows and congestion". Technical Report. Department of Information and System Engineering, Faculty of Science and Engineering, Chuo University, TRISE 94-03, 1994.

[9] D. Pham and T. Le Thi. "Convex analysis approach to d. c. programming theory, algorithms and applications". Acta Mathematica Vietnamica, Vol. 22, pp. 289-355, 1997.

[10] D. Pham. "Duality in d. c. (difference of convex functions) optimization. Sub-gradient methods". Trends in Mathematical Optimization, International Series of Numerical Mathematics 84, Birkhauser pp. 277-293, 1988.

[11] J. Phillips. "Algorithms for the vector maximization problem". Mathematical Programming, Vol. 2, pp. 207-229, 1972.

[12] J. Shi and Y. Yamamoto. "A global optimization method for minimum maximal flow problem". ACTA Mathematica Vol., 22, pp. 271-287, 1997.

[13] M. Shigeno, I. Takahashi and Y. Yamamoto. "Minimum maximal flow problem: an optimization over the efficient set". Journal of Global Optimization, Vol. 25, pp. 425-443, 2003.

[14] Y. Yamamoto. "Optimization over the efficient set, Overview". Journal of Global Optimization, Vol. 1-4, pp. 285-317, 2001.

[15] D. Tuyttens, J. Teghem, N. El-Sherbeny. (2004). "A Particular Multiobjective Vehicle Routing Problem Solved by Simulated Annealing". Lecture Notes in Economics and Mathematical Systems, Vol. 535, pp. 133-152, Springer-Verlag, Berlin, Germany. Available: http://dx.doi.org/10.1007/978-3-642-17144-4