Maximum Flow of Minimum Bi-criteria Cost in Dynamic Networks

MIRCEA PARPALEA\(^1\), ELEONOR CIUREA\(^2\)

\(^1\)National College “Andrei Saguna”
1, Saguna, street, 500123, Brasov
\(^2\)Department of Computer Science
Transilvania University of Brasov
25, Eroilor, blvd, 500030, Brasov

ROMANIA

e.ciurea@unitbv.ro, parpalea@gmail.com, http://www.unitbv.ro

Abstract: The article studies the generalisation of the maximum flow of minimum cost problem for the case of maximum discrete dynamic flow of minimum travelling cost and travelling time. The approach is based on iteratively generating efficient extreme points in the objective space by solving a series of single objective maximum flow problems with different objective functions. On each of the iterations, the flow is augmented along a cheapest path from the source node to the sink node in the time-space network avoiding the explicit time expansion of the network.

Key-Words: Dynamic network, maximum flow, bi-criteria flow, successive shortest path

1 Introduction

Classical (static) network flow models have been well known as valuable tools for many applications. However, they fail to capture the dynamic property of many real-life problems, such as traffic planning, production and distribution systems, communication systems, and evacuation planning. In transportation problems or in network flows problems, the selection of the optimum solution takes into account more than one criterion, e.g. minimization of the cost for selected routes, the minimization of arrival time at the destinations, the minimization of the deterioration of goods, the minimization of the load capacity, the maximization of safety, reliability, etc. Often, these criteria are in conflict and for this reason, a multi-objective network flow formulation of the problem is necessary.

2 Terminology and preliminaries

2.1. Dynamic network flows

A discrete dynamic network \(G=(N,A,T)\) is a directed graph where \(N=\{\ldots,i,\ldots\}\) is a set of nodes \(i\) with \(|N|=n\), \(A=\{\ldots,a,\ldots\}\) is a set of arcs \(a\) with \(|A|=m\), and \(T\) is a finite time horizon discretized into the set \(\{0,1,\ldots,T\}\). An arc \(a\) from node \(i\) to node \(j\) is usually also denoted by \((i,j)\). The following functions are associated with each arc \(a=(i,j)\) in \(A\):
- the time-dependent capacity (upper bound) function \(u(i,j;\theta)\), \(u: A \times \{0,1,\ldots,T\} \rightarrow \mathbb{R}^+\) which represents the maximum amount of flow that can enter the arc \((i,j)\) at time \(\theta\), the time-dependent transit time function \(h(i,j;\theta)\), \(h: A \times \{0,1,\ldots,T\} \rightarrow \mathbb{R}\), and the time-dependent cost function \(c(i,j;\theta)\), \(c:A \times \{0,1,\ldots,T\} \rightarrow \mathbb{R}\) which represents the cost for sending one unit of flow through the arc \((i,j)\) at time \(\theta\). The time horizon \(T\) is the time until which the flow can travel in the network.
- The demand-supply function \(v(i;\theta)\), \(v: N \times \{0,1,\ldots,T\} \rightarrow \mathbb{R}\) represents the demand of node \(i\) at the time-moment \(\theta\) if \(v(i;\theta)<0\) or the supply of node \(i\) at the time-moment \(\theta\) if \(v(i;\theta)>0\). The network has two special nodes: a source node \(s\) with \(v(s;\theta)\geq 0\) for \(\theta\) in \(\{0,1,\ldots,T\}\) and there exists at least one moment of time \(\theta_i\) in \(\{0,1,\ldots,T\}\) such that \(v(s;\theta_i)\geq 0\); and a sink node \(t\) with \(v(t;\theta)\leq 0\) for \(\theta\) in \(\{0,1,\ldots,T\}\) and there exists at least one moment of time \(\theta_i\) in \(\{0,1,\ldots,T\}\) such that \(v(t;\theta_i)\leq 0\). The condition required for the flow to exist it that \(\sum_{\theta\in[0,1,\ldots,T]} \sum_{i\in N} v(i;\theta) = 0\).

A feasible dynamic flow \(f(i,j;\theta)\) (flow over time) on \(G=(N,A,u,h,c,T)\) with time horizon \(T\) is a function \(f: A \times \{0,1,\ldots,T\} \rightarrow \mathbb{R}\) that satisfies the following flow conservation constraints \(\forall \theta \in [0,1,\ldots,T]\):
\[ \sum_{j \in A(i)} f(i,j;\theta) - \sum_{h(j,i;\theta)} h(j,i;\theta) = v(i;\theta), \quad (1.a) \]

\[ \forall i \in N; \text{ where } f(i,j;\theta) \text{ determines the rate of flow (per time unit) entering arc } (i,j) \text{ at time } \theta. \]

Capacity constraints (1.b) mean that in a feasible dynamic flow, at most \( u(i,j;\theta) \) units of flow can enter the arc \((i,j)\) at the time-moment \( \theta \).

\[ 0 \leq f(i,j;\theta) \leq u(i,j;\theta), \quad \forall \theta \in [0,1,\ldots,T]; \quad (1.b) \]

\[ f(i,j;\theta) = 0, \quad \theta \in T - h(i,j;\theta) + 1, T. \quad (1.c) \]

\[ \forall (i,j) \in A; \]

It is easy to observe that the flow does not enter arc \((i,j)\) at time \( \theta \) if it has to leave the arc after time \( T \); this is ensured by condition (1.c). The total cost of the dynamic flow \( f(i,j;\theta) \) in a dynamic network is defined as:

\[ C(f) = \sum_{\theta \in [0,1,\ldots,T]} \sum_{(i,j) \in A} f(i,j;\theta) \cdot c(i,j;\theta) \quad (2) \]

2.2. Time-space network

In the discrete time model, a useful tool for studying the minimum cost flow over time problem is the time-space network. The time-space network is a static network constructed by expanding the original network in the time dimension by considering a separate copy of every node \( i \in N \) at every time step in the time horizon \( T, \theta \in [0,1,\ldots,T] \).

A node-time pair (NTP) \((i,\theta)\) refers to a particular node \( i \in N \) at a particular time step \( \theta \in [0,1,\ldots,T] \), i.e., \((i,\theta) \in N \times [0,1,\ldots,T]\).

The NTP \((i,\theta)\) is linked to the NTP \((j,\theta_2)\) if either

(i) \((i,j) \in A \) and \( \theta_2 = \theta + h(i,j;\theta) \), or

(ii) \((i,j) \in A \) and \( \theta_1 = \theta + h(j,i;\theta_2) \).

**Definition 1:** The time-space network \( G^T \) of the original dynamic network \( G \) is defined as follows:

\[ N^T := \{(i,\theta)|i \in N, \theta \in [0,1,\ldots,T]\}; \quad (3.a) \]

\[ A^T := \{a = (i,\theta,(j,\theta+h(i,j)))|(i,j) \in A, 0 \leq \theta \leq T-h(i,j)\}; \quad (3.b) \]

\[ u^T(a) := u(a) \quad \text{for} \quad a \in A^T; \quad (3.c) \]

\[ c^T(a) := c(a) \quad \text{for} \quad a \in A^T. \quad (3.d) \]

For every arc \((i,j) \in A \) with traversal time \( h(i,j) \), capacity \( u(i,j) \), and cost \( c(i,j) \), the time-space network \( G^T \) contains arcs \((i,\theta),(j,\theta+h(i,j))\) for \( \theta = 0,1,\ldots,T-h(i,j) \) with capacities \( u(i,j) \) and costs \( c(i,j) \). For the flow \( f(a;\theta) \) in the dynamic network \( G \), the function \( f^T(a_\theta) \) that represents the corresponding flow in the time-space network \( G^T \) is defined as:

\[ f^T(a_\theta) = f(a;\theta), \quad \forall a_\theta \in A^T. \quad (4) \]

A dynamic path is defined as a sequence of distinct, consecutively linked NTPs:

\[ P(i,j): (i,\theta) = (i_1,\theta_1),(i_2,\theta_2),\ldots,(i_r,\theta_r) = (j,\theta). \quad (5) \]

2.3. Time-dependent residual network

The time-dependent residual network corresponding to a feasible flow \( f \) can be viewed as the static residual network of the time-space network corresponding to the dynamic network. For \( f(i,j;\theta) \) being the flow entering arc \((i,j)\) at time \( \theta \), an additional flow \( u(i,j;\theta) - f(i,j;\theta) \) departing from node \( i \) at time \( \theta \) to node \( j \) along the arc \((i,j)\) can be sent. Also, \( f(i,j;\theta) \) units of flow can be sent from node \( j \) departing at time \( \theta + h(i,j;\theta) \) and consequently arriving at node \( i \) at time \( \theta \) over the arc \((i,j)\), which amounts to cancelling the existing flow on the arc. Here, an arc with negative travel time (i.e., departing at \( \theta + h(i,j;\theta) \) and arriving at \( \theta ) \) is considered. Whereas sending a unit of flow from \( i \) at time \( \theta \) to \( j \) along \((i,j)\) increases the flow cost by \( c(i,j;\theta) \) units, sending a unit of flow in reverse direction from \( j \) departing at time \( \theta + h(i,j;\theta) \) to \( i \) on the same arc decreases the flow cost by \( c(i,j;\theta) \) units. Considering the above mentioned ideas, the residual network with respect to a current dynamic flow \( f \) is defined as follows.

**Definition 2:** The residual dynamic network with respect to a given feasible dynamic flow \( f \) is defined as \( G(f) = (N, A(f), T) \) with \( A(f) := A^*(f) \cup A'(f) \) where \( A(f) := \{(i,j)|i \in A, \exists \theta \leq T - h(i,j;\theta) \} \)

\[ \text{with} \quad u(i,j;\theta) - f(i,j;\theta) > 0 \quad (6.a) \]

and \( A'(f) := \{(i,j)|i \in A, \exists \theta \leq T - h(i,j;\theta) \}

\[ \text{with} \quad f(i,j;\theta) > 0 \quad (6.b) \]

While the direct arcs \((i,j) \in A(f)\) have the same transit times \( h(i,j;\theta) \) and costs \( c(i,j;\theta) \) as in the original dynamic network \( G \), the artificial reverse arcs \((i,j) \in A'(f)\) in the residual dynamic network \( G(f) \) are provided with the following attributes:

\[ h(i,j;\theta) + h(j,i;\theta) := -h(j,i;\theta), \quad (7) \]
Recent Researches in Computer Science
ISBN: 978-1-61804-019-0

\[ c(i, j; \theta + h(j, i; \theta)) := -c(j, i; \theta), \]  
for \((i, j) \in A, 0 \leq \theta + h(j, i; \theta) \leq T, f(j, i; \theta) > 0\).

The residual capacities of the arcs \((i, j)\) in the residual dynamic network \(G(f)\) are defined as follows:

\[ r(i, j; \theta) := u(i, j; \theta) - f(i, j; \theta), \]
\((i, j) \in A, 0 \leq \theta + h(j, i; \theta) \leq T\) \hspace{1cm} (9.a)

\[ r(i, j; \theta + h(j, i; \theta)) := f(j, i; \theta), \]
\((i, j) \in A, 0 \leq \theta + h(j, i; \theta) \leq T\) \hspace{1cm} (9.b)

**Definition 3:** A dynamic path \(P(s = i_1, i_2, \ldots, i_q = i)\) from node \(s\) to node \(i\) is said to be a dynamic augmenting path if \(r(i_k, i_{k+1}; \theta_k) > 0\) for \((i_k, i_{k+1}) \in A(f)\) and \(k = 1, \ldots, q - 1\).

**Definition 4:** Given a dynamic flow \(f\), the residual capacity of a dynamic augmenting path \(P(s = i_1, i_2, \ldots, i_q = i)\) is defined by:

\[ r(P) := \min_{1 \leq k \leq q - 1} r(i_k, i_{k+1}; \theta_k) \]
for \((i_k, i_{k+1}) \in A(f)\) and \(k = 1, \ldots, q - 1\). \hspace{1cm} (10)

**Definition 5:** The cost of a dynamic augmenting path \(P(s = i_1, i_2, \ldots, i_q = i)\) is defined by:

\[ C(P) := \sum_{(i_k, i_{k+1}) \in A(f)} c(i_k, i_{k+1}; \theta_k) \]
for \(k = 1, \ldots, q - 1\).

A dynamic augmenting path \(P(s = i_1, i_2, \ldots, i_q = i)\) is referred to as a dynamic shortest augmenting path (DSAP) from node \(s = i_1\) to node \(i_q = i\) if \(C(P) \leq C(P')\) for all dynamic augmenting paths \(P'\) from node \(s\) to node \(i\).

A dynamic path \(P(i, i_2, \ldots, i_q; \theta_2, \ldots, \theta_q)\) is called a dynamic cycle if \(i_q = i_1\) and \(\theta_q = \theta_2\). A negative cycle is defined as a dynamic cycle whose total cost is negative and whose capacity is greater than zero.

### 2.3. The Maximum Dynamic Flow – Minimum Dynamic Cut Theorem

A dynamic cut is considered as a dynamic extension of the static cut.

**Definition 6:** A node \(j\) is said to be reachable from another node \(i\) if there exists a dynamic augmenting path from \(i\) to \(j\).

Considering two set-valued functions \(S(\theta)\) and \(\overline{S}(\theta) = N - S(\theta)\) for all \(\theta \in [0, 1, \ldots, T]\), the collection of all \(S(\theta)\) is referred to as a generalized cut of separating node \(s\) and \(t\) and is defined as follows:

**Definition 7:** The generalized \(s-t\) dynamic cut \(S\) is a set of valued function of time defined as:

\[ S = \{ S(\theta) | S(\theta) \subset N, s \in S(\theta), t \notin S(\theta), \theta = 0, 1, \ldots, T \} \]

**Definition 8:** The capacity of the generalized \(s-t\) dynamic cut \(S\) is defined as:

\[ \text{Cap}(S) := \sum_{i \in S, j \notin S} u(i, j; \theta) \]

The minimum \(s-t\) dynamic cut is the \(s-t\) dynamic cut having the minimum value of it's capacity among all \(s-t\) dynamic cuts.

**Theorem 1:** Let \(v\) be the value of any feasible dynamic flow \(f\) in \(G = (N, A, T)\) and \(\text{Cap}(S)\) be the value of any generalized cut \(S\). Then, \(v \leq \text{Cap}(S)\). \hspace{1cm} (see [2])

**Theorem 2** (Maximum Dynamic Flow - Minimum Dynamic Cut): The value of the maximum dynamic flow from a source node \(s\) to the sink node \(t\) equals the value of minimum \(s-t\) dynamic cut. \hspace{1cm} (see [6])

### 3. Successive Shortest Path Algorithm for Dynamic Max Flow of Minimum Cost

#### 3.1. Dynamic Shortest Paths

Solution approaches for classical shortest path problems are divided into two classes: label-setting and label-correcting [5]. The residual time-space network is composed of two sub-networks: a forward network consisting of the set of forward arcs, denoted by \(A^+(f)\), having positive travel times and travel costs; and a reverse network consisting of the set of reverse arcs, denoted by \(A^-(f)\) and having negative travel times and travel costs. Each of the two sub-networks, alone, is acyclic.

Our approach consists in exploring the forward and reverse arcs simultaneously. For every augmentation, a set \(L\) of candidate nodes is maintained, which initially includes only the source node. The set \(L\) holds all node-time pairs which have been reached so far by the algorithm and which are to be visited. The minimum cost labels \(\pi(i, \theta)\) of all node-time pairs are initialised to infinity with the exception of the minimum cost labels of the source node which are initialised to zero, \(\pi(s, \theta) := 0, \forall \theta \in [0, 1, \ldots, T]\).

For every node-time pair \((i, \theta)\) selected from \(L\), the arcs with positive residual capacity connecting \((i, \theta)\) to \((j, \theta)\) are explored, where
0 < \theta = \theta + h(i,j; t) \leq T \text{ if the arc connecting } (i, \theta) \text{ to } (j, \theta) \text{ is a forward arc and } 0 \leq \theta = \theta - h(j,i; t) \leq T \text{ if it is a reverse arc. Then the minimum cost labels are updated and the node-time pair } (j, \theta) \text{ is added to the candidate set if it is not already in } L. \text{ The process is repeated until there are no more candidate nodes in } L. \text{ The travel cost of the minimum cost path computed based on predecessor vector } p \text{ is given by } \pi(t) = \min_{\theta \in [0, 1, \ldots, T]} \{\pi(t, \theta)\}.

The Dynamic Shortest Path (DSP) procedure is presented in Figure 1.

![Figure 1. Procedure Dynamic Shortest Path (DSP)](image)

Cai, X. et al [2] proved that the complexity of finding a shortest dynamic flow-augmenting path, by exploring the two sub-networks successively, is \(O(mnT^2)\). For algorithms which explores the forward and reverse arcs simultaneously, Miller-Hooks and Patterson [3] reported a complexity of \(O(n^2T^2)\).

By using special node addition and selection procedures, Nasrabadi and Hashemi [4] succeeded to reduce significantly the number of node time pair that needs to be visited. The worst-case complexity of their algorithm is \(O(n(T(n + T)))\).

### 3.2. Successive shortest path algorithm

The successive shortest path algorithm for finding a maximum flow of minimum cost will repeatedly perform the following operations:

1. Compute a minimum cost path \(P\) from the source node to the sink node;
2. Find the residual capacity \(r(P)\) of the minimum cost path;
3. Augment the flow along the minimum cost path and update the residual network;

The algorithm in Figure 2 will terminate when none of the sink node-time pairs \((t, \theta)\), \(\forall \theta \in [0, 1, \ldots, T]\) is reachable from any of the source node-time pairs \((s, \theta)\), \(\forall \theta \in [0, 1, \ldots, T]\) which represent that there is no feasible dynamic flow augmenting path from \(s\) to \(t\).

![Figure 2. Procedure Successive Shortest Path (SSP)](image)

**Theorem 3:** Procedure Successive Shortest Path (SSP) computes correctly the maximum dynamic flow of minimum cost for a given time horizon \(T\).

**Proof:** The procedure terminates when the sink node is not reachable from the source node, i.e. there does not exist a dynamic augmenting path from the source node to the sink node in the time-depending residual network, meaning that a maximum flow is obtained. Since in every step the augmentation is performed over the current minimum cost path, the obtained flow is also a minimum cost flow.

Denoting by \(\bar{u}\) the maximum value for the upper bounds of all arcs, the following theorem can be formulated:
Theorem 4: Procedure Successive Shortest Path (SSP) can be implemented in $O(nmnT^2)$ time.

Proof: For the labelling operation, all arcs at all times may be examined, so the running time is $O(mT)$. Updating the residual networks also requires a running time of $O(mT)$, hence the complexity of one iteration is bounded by $O(mT)$. Since at each time $0 \leq t \leq T$ there may be no more than $n$ paths sending flow to the sink node $t$ and the maximum flow on any possible path is at most $\bar{\pi}$, the maximum flow value is bounded by $O(nT\bar{\pi})$. Considering that each iteration at least augments one unit of flow, i.e. the algorithm terminates in $nT\bar{\pi}$ iterations, the total running time is bounded by $O(nmnT^2)$.

4 Bi-criteria minimum cost maximum dynamic flow

4.1. Problem formulation

Let us consider a discrete-time dynamic network flow problem which searches for the optimum solution by taking into account more than one criterion. By setting the two objective functions to minimizing the travelling time and the travelling cost, the bi-criteria problem of finding the maximum dynamic flow of minimum travel time and travel cost, the bih criteria problem of finding the maximum cost can be formulated as follows:

Maximize $\sum_{\theta=0}^{T} \sum_{(i,j) \in A} \sum_{b \in b(i,j;\theta)} f(i,t;\theta)$ (11.a)

Minimize $y_1(f) = \sum_{\theta=0}^{T} \sum_{(i,j) \in A} c(i,j;\theta) \cdot f(i,j;\theta)$ (11.b)

Minimize $y_2(f) = \sum_{\theta=0}^{T} \sum_{(i,j) \in A} h(i,j;\theta) \cdot f(i,j;\theta)$ (11.c)

subject to:

$\sum_{b \in b(i,j;\theta)} f(i,j;\theta) - \sum_{[\xi,\eta] \in [i,j],\theta \in \Theta} f(\xi,\eta;\theta) = 0 \forall i \in N - \{s,t\}$ (11.d)

$0 \leq f(i,j;\theta) \leq u(i,j;\theta)$, $\forall \theta \in [0,1,\ldots,T], \forall (i,j) \in A$ (11.e)

Here, the value of the maximum dynamic flow for a time horizon $T$ is denoted by $v$ where $f$ is the vector of flow on arcs. Any vector $f$ that satisfies the flow conservation constraint (11.d) at the different node-time pairs and the bound constraint (11.e) is called a feasible solution of the discrete dynamic maximum flow of bi-criteria minimum cost (BiMCMDF) problem.

The set of feasible solutions or decision space is denoted by $F$ and its image through $Y(F) = \{y_1(f),y_2(f)\} | f \in F$ is called objective space. In general, there is no feasible solution of the BiMCMDF problem that simultaneously minimizes both objectives. In other words, an optimum global solution does not exist. For this reason, the solutions of these problems are searched for among the set of efficient points.

Definition 9: A feasible solution $f \in F$ of the bi-criteria minimum cost flow problem is called efficient if, and only if, there does not exist another feasible solution $f' \in F$ so that $Y(f') \leq Y(f)$ with $Y(f') \neq Y(f)$ (i.e. $y_1(f') \leq y_1(f)$, with at least one strict inequality, $k \in \{c,h\}$).

Definition 10: $Y(f)$ is a non-dominated criterion vector if $f$ is an efficient solution. Otherwise $Y(f)$ is a dominated criterion vector.

The set of efficient solutions of $F$ will be denoted by $E[F]$ while, by extension, $E[Y(F)]$ is called the set of non-dominated solutions of $Y(F)$. It is well known that to characterize $E[Y(F)]$ for the bi-criteria continuous minimum cost flow problem, it is only necessary to identify the extreme efficient points of $Y(F)$. The set of efficient extreme points of $F$ will be denoted by $F^\ast[F]$ and the corresponding points of $Y(F)$ will be denoted by $Y^\ast[F]$. The set of non-extreme efficient points on the efficient boundary of $F$ will be denoted by $F_{na}[F]$ and the corresponding set in the objective space by $Y_{na}[Y(F)]$. In the BiMCMDF problem all the efficient solutions lie on the efficient boundary of $Y(F)$.

4.2. The algorithm

Aneja and Nair [1] developed a simple algorithm for bi-criteria transportation problems. Their procedure generates efficient extreme points on the objective space $Y(F)$ rather than on the decision space $F$.

A series of single objective problems are solved with different objective functions and each problems leads to either a new efficient extreme point or changes the direction of search in the objective space. The algorithm terminates when no extreme point or no improving direction is available. Let $Y_1$ and $Y_2$ be two efficient extreme points in the objective space $Y(F)$ which correspond to the efficient solutions $f_1$ and $f_2$ obtained by solving the two single objective problems $\min y_1(f)$ and

\[ \min y_1(f) \text{ subject to } y_2(f) \leq \beta \]

\[ \min y_2(f) \text{ subject to } y_1(f) \leq \beta \]

where $\beta$ is the current value of $\beta$.
$\min y_i(f)$ respectively. Setting $\alpha := y_i(f_i) - y_i(f_l)$ and $\beta := y_i(f_h) - y_i(f_l)$, the slope between the two efficient extreme points in the objective space $Y_i$ and $Y_j$ is $\lambda = -\beta / \alpha$. New artificial costs $c'(i, j; \theta) := \beta \cdot c(i, j; \theta) + \alpha \cdot h(i, j; \theta)$ are computed for all the arcs in the network and another single objective problem with the objective function

$$y(f) = \sum_{(i, j) \in A} \sum_{(i, j; \theta)} c'(i, j; \theta) \cdot f(i, j; \theta)$$

is solved.

Theorem 5: The Bi-criteria Minimum Cost Maximum Dynamic Flow (BiMCMDF) algorithm computes the set of extreme non-dominated points in the objective space in $O(K \cdot T_{nn} T^3)$ time.

Proof: The proof results directly from Theorem 4. ■

5 Conclusion

Our approach to the problem of the maximum flow of Bi-criteria cost in dynamic networks considered the case of discrete dynamic network for which the two criteria taken in account consist in minimizing the travelling time and the travelling cost for a maximum possible flow which can be sent from a source to a sink within a time horizon $T$. The proposed method iteratively generates efficient extreme points in the objective space by solving a series of single objective maximum flow problems with different objective functions. On each of the iterations, the flow was augmented along a cheapest path from the source node to the sink node in the time-space network, avoiding the explicit time expansion of the network.

References: