ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **23**, Number T, 2020, T18-T27

Maximum Flows in Planar Dynamic Networks. The Static Approach

Camelia SCHIOPU[1], Eleonor CIUREA[2]
[1] Transilvania University of Braşov, B-ul Eroilor, 29, 500036 Braşov, Romania E-mail: camelia.s@unitbv.ro
[2] Department of Mathematics and Computer Science, Transilvania University of Braşov, Iuliu Maniu, 50, 500091 Braşov, Romania
E-mail: e.ciurea@unitbv.ro

Abstract. The The research of flows in planar static networks is motivated by the fact that more efficient algorithms can be developed by exploiting the planar structure of the graph. An nontrivial extension of the maximum static flow problem is the maximum dynamic flow model, where the transit time to traverse an arc is taken into consideration. The problem of finding a maximum flow in dynamic network is more complex than the problem of finding a maximum flow in static network. Happily, this complication can be solved by rephrasing the problem from dynamic network D in a multiple source multiple sink static reduced expanded network R_0. If dynamic network D is planar, we show that the network R_0 is planar.

Keywords: planar network, dynamic network, maximum flow

1. Introduction

The maximum flow in static network with zero lower bounds aims to find a solution that can send the maximum flow from a set of distinguished nodes, called sources, to another set of distinguished nodes, called sinks, under the constraint that the zero lower bounds and the upper bounds (capacities) of the arcs in the network are all satisfied. Many efficient algorithms have been developed to solve the maximum flow problem in static general network [1]. In this case, the problem of maximum flow with multiple sources and sinks can be reduced to the single source single sink by introducing an artificial source and an artificial sink and connecting them to all the sources and sinks, respectively [1].

The static network models provide good mathematical representation of many applications. In some other applications, the time is an essential ingredient. In these instances we need to use dynamic network flow models. For dynamic flow problem see [1], [2], [3] and other works.

The maximum flow in the planar static networks problem has many applications and hence it is of interest to find fast flow algorithms for this class of networks. This problem starting with Ford and Fulkerson [4] who gave a particular augmenting path algorithm for the case of a planar graph where source node 1 and sink node n are on the same face. Many researchers have investigate the computation of maximum flow for following cases:

(a) planar graphs where source node and sink node are on the same face;

(b) undirected planar graphs where source node and sink node are on the arbitrary faces;

(c) directed planar graphs where source node and sink node are on the arbitrary faces;

(d) planar graphs with multiple sources and sinks, when it is known how much of the commodity is produced/consumed at each source and each sink;

(e) planar graphs with multiple sources and sinks, when all sources and sinks are on the boundary of a single face;

(f) planar graphs with multiple sources and sinks, when the sources and the sinks reside on the boundaries of different faces.

This problems have investigated in works [4], [5], [6], [7], [8], [9], [10], [11], and other works.

The dynamic approach of maximum flow in $(1, n)$ planar dynamic networks with zero lower bounds and upper bounds positive have investigate in paper [12], where 1 is source node and n is sink node.

In this paper we present the static approach of maximum flow in $(1, n)$ planar dynamic networks with zero lower bounds and upper bounds (capacities) positive. We present the case when the planar dynamic network has the transit times $h(i, j; t) = h(i, j)$ for all $(i, j) \in A$. Further on, in Section 2 some basic dynamic network notations and terminology are presented. In Section 3 the maximum flow in directed planar static networks with multiple sources and sinks is exposed, while in Section 4 is presented the maximum flow in directed $(1, n)$ planar dynamic network. In Section 5 an example is given.

The maximum flow problem in static networks arise in a wide variety of situations and in several forms. The maximum flow problem arises in a number of combinatorial applications that on the surface with not appear to be maximum flow problem at all. The problem also arises directly in problems as far reaching as machine scheduling, the assignment of computer modules to computer processors, the rounding of census data to retain the confidentiality of individual households, and several other applications.

Dynamic network models arise in many problem settings, including production distribution system, economic planning, energy systems, traffic systems, building evacuation systems, and several other applications.

In many of this applications the networks are planar.

In the following presentation, some familiarity with maximum flow problem are assumed and many details are omitted, since the result are known and can be consulted in references.

The problem of maximum flow in dynamic networks with the static approach is treated so far only by the authors of this paper.

2. Maximum flow in planar static networks with multiple sources and sinks

Let $G = (N, A, u)$ be a general static network with the set of nodes $N = \{1, \dots, i, \dots, j, \dots, n\}$, the set of arcs $A = \{a_1, \dots, a_k, \dots, a_m\}$, $a_k = (i, j)$ and upper bound (capacity) function $u: A \to \mathbb{R}_+$, where $\mathbb{R}$ is real number set. To define maximum flow in static networks with multiple sources and sinks, we distinguish two special sets of nodes in the network G: a set of source nodes $N_1 = \{1, 2, \dots, n_1\}$, a set of transit nodes $N_2 = \{n_1 + 1, n_1 + 2, \dots, n_2\}$ and a set of sink nodes $N_3 = N - (N_1 \cup N_2) = \{n_2 + 1, n_2 + 2, \dots, n_3\}$ with $n_3 = n$.

A static flow in network G is a function $f: A \to \mathbb{R}_+$, satisfying the following conditions:

$$\sum_j f(i,j) - \sum_k f(k,i) = \begin{cases} v(i), & if\ i \in N_1 & (1a) \\ 0, & if\ i \in N_2 & (1b) \\ -v(i), & if\ i \in N_3 & (1c) \end{cases}$$

with $\sum_{N_1} v(i) - \sum_{N_3} v(i) = 0$. The value $v = \sum_{N_1} v(i) = \sum_{N_3} v(i)$ is the value of flow f.

A feasible static flow in network G is a flow in G which satisfies the following conditions:

$$0 \leq f(i,j) \leq u(i,j)\ for\ all\ (i,j) \in A \tag{2}$$

The maximum flow problem in network G is to determine a feasible flow which maximizes v. A cut in network G is a partition of the node set N into two subset S and $\bar{S} = N - S$. We represent this cut using notation $[S, \bar{S}]$. We refer to an arc (i, j) with $i \in S$ and $j \in \bar{S}$ as a forward arc of the cut and an arc (i, j) with $i \in \bar{S}$ and $j \in S$ as a backward arc of the cut. Let $(S, \bar{S})$ denote the set of forward arcs in the cut and let $(\bar{S}, S)$ denote the set of backward arcs. We have that the arc set of cut is $[S, \bar{S}] = (S, \bar{S}) \cup (\bar{S}, S)$. We refer to a cut $[S, \bar{S}]$ as an $N_1 - N_3$ cut if $N_1 \subseteq S$ and $N_3 \subseteq \bar{S}$.

For the maximum flow problem (1), (2), we define the capacity of the $N_1 - N_3$ cut $[S, \bar{S}]$ as:

$$c[S, \bar{S}] = u(S, \bar{S}) = \sum_{(S, \bar{S})} u(i,j) \tag{3}$$

We refer to an $N_1 - N_3$ cut whose capacity is the minimum among all $N_1 - N_3$ cuts as a minimum cut. Recall the maximum flow and minimum cut theorem [1].

Theorem 1. *The maximum value of the flow from the set sources N_1 to the set sinks N_3 in network G equals the capacity of minimum $N_1 - N_3$ cut.*

We remark the fact that the maximum flow problem into general network with multiple sources and sinks is equivalent with a maximum flow problem into extended general network with a single source and a single sink by introducing an artificial source and an artificial sink and connecting them to all the sources and sinks, respectively [1], [4].

The concept of residual network plays a central role in the development of all the maximum flow algorithms [1]. If the network G contains arc (i, j) and not contains arc (j, i) we consider that G also contains arc (j, i) with $u(j, i) = 0$. The residual capacity $r(i, j)$ for any arc $(i, j) \in A$ is $r(i, j) = u(i, j) - f(i, j) + f(j, i)$. The residual network $\tilde{G} = (N, \tilde{A}, r)$ has $\tilde{A} = \{(i, j) | (i, j) \in A, r(i, j) > 0\}$.

Next we consider that the network G is a planar network. First, we present some definitions and properties of planar digraphs (directed graphs) $G = (N, A)$ [1].

Definition 1. *A digraph $G = (N, A)$ is said to be planar if we can draw it in a two dimensional plane so that no two arcs intersect each other.*

For more precisely of this definition see other references of this paper.

Researchers have developed very efficient algorithms (in fact, linear time algorithms) for testing the planarity of a digraph. A much simpler but nevertheless fairly efficient algorithm is presented in paper [3]. The result of this algorithm is a planar drawing of G, or decides that digraph G is non planar.

Next, we describe the algorithm due to Democron, Malgrange et Pertuiset [3].

A digraph $G = (N, A)$ is planar if it is isomorphic to a digraph $G' = (N', A')$ such that the N' and A' are contained in the same plan and such that at most one node occupies or at most one are passes through any point of the plane. G' is said to be embedded in the plane and to be a planar representation of G. In general, $\tilde{G}$ will denote an embedding of G.

By $G_1 = (N_1, A_1)$ we denote a subgraph of $G = (N, A)$. A piece of G relative to G_1 is then:

a) an arc $(i, j) \in A$ where $(i, j) \in A$ where $(i, j) \in A_1$ and $i, j \in N_1$,

or

b) a connected component of $(G - G_1)$ plus any arcs incident with this component.

If a piece has two or more points of contact then it is called a bridge.

Let C be any circuit which is a subdigraph of G. $\tilde{C}$ then divides the plane in two faces, an inner and an outer face. For every pair of nodes of a given bridge of C, there is a path from one node to another which does not use an arc of C.

Let $\tilde{G}_k$ be a planar embedding of the subdigraph of G. If there exists a planar embedding $\tilde{G}$, such that $\tilde{G}_k \subseteq \tilde{G}$, then $\tilde{G}_k$ is said to be G-admissible. Let B be any bridge of G relative to G_k. Now, B can be drawn in a face of $\tilde{G}_k$ if all the points of contact of B are in the boundary of n'. By $F(B, \tilde{G}_k)$ we denote the set of faces F of $\tilde{G}_k$ in which B is drawable.

The planarity testing algorithm is outline in Algorithm 1. The algorithm finds a sequence of digraphs $G_1, G_2, \ldots$, such that $\tilde{G}_k \subset \tilde{G}_{k+1}$ and finds their planar embeddings $\tilde{G}_1, \tilde{G}_2, \ldots$. If G is planar then, as shall see, each $\tilde{G}_i$ found by the algorithm is G-admissible and the algorithm terminates with a planar embedding of G, $\tilde{G}_{m-n+2}$. If G is non-planar then the algorithm stops with the discovery of some bridge B (with respect to the current $\tilde{G}_k$) for which $F(B, \tilde{G}_k) = \emptyset$. Obviously a necessary condition that $\tilde{G}_k$ is G-admissible is that for every bridge B relative to $\tilde{G}_k$, $F(B, \tilde{G}_k) \neq \emptyset$.

Algorithm 1: A planarity testing algorithm

Find a circuit c of G
$G_1 := C$; $\tilde{G}_1 := C$;
$n' := 2$;
$b := true$;
while $n' \neq m - n + 2$ and $b = true$ **do**
    find each bridge B of G relative to G_k;
    for each B **do**
        find $F(B, \tilde{G}_k)$;
    end for;
    if for some B, $F(B, \tilde{G}_k) \neq \emptyset$ then
        $b := false$; output the message 'G is non-planar';
    end if;
    if $b = true$ **then**
        if for some B $|F(B, \tilde{G}_k)| = 1$ **then**
            $F := F(B, \tilde{G}_k)$;
        else
            let B be any bridge and F be any face such that $F \in F(B, \tilde{G}_k)$;
        end if;
        find a path $P_k \subseteq B$ connecting two points of contact of B to $\tilde{G}_k$;
        $G_{k+1} := G_k + P_k$;
        Obtain a planar embedding $\tilde{G}_{k+1}$ of G_{k+1} by drawing P_k in the face F of $\tilde{G}_k$;
        $k := k + 1$; $n' := n' + 1$;
        if $n' := m - n + 2$ **then**
            output the message 'G is planar';
        end if;
    end if;
end while.

Definition 2. *A bordered face of planar digraph G is a region of the plane bounded by arcs that satisfies the condition that any two points in the region can be connected by a continuous curve that meets no nodes and arcs. The boundary of a bordered face is the set of all arcs that enclose it. Two faces said to be adjacent if their boundaries contain a common arc.*

We specify that the planar digraph G has an unbounded face.

Theorem 2. *If a connected planar digraph has n nodes, m arcs and n' faces, then* $n' = m - n + 2$.

Theorem 3. *If a planar digraph has n nodes and m arcs, then $m < 3n$.*

Now, we define the dual directed static network denoted by $G' = (N', A', u')$. The set $N = \{1', \ldots, i', \ldots, j', \ldots, n'\}$ is the set of nodes, a node for each face of G. The set $A' = \{a'_1, \ldots, a'_k, \ldots, a'_m\}$ is the set of arcs. There is an one to one correspondence between A and A' as follows: for each arc $a_k \in A$, let a'_k be the corresponding dual arc connecting the right face bordering a_k with the left face bordering on a_k.

We use the clockwise rule: if the arc $a_k = (i, j)$ is a clockwise arc in the face i', we define the dual arc $a'_k = (i', j')$. This rule is also known as the left hand rule: if the thumb points in the direction of a_k, then the index finger points in the direction of a'_k. The weight of a_k arc is $u'(a'_k) = u(a_k)$.

A potential function on a network $G = (N, A, u)$ is $p: N \to \mathbb{R}$. The potential function p is called feasible (consistent) if $p(j) - p(i) \leq u(i, j)$ for all $(i, j) \in A$. Given a feasible potential function p defined on dual network $G' = (N', A', u')$, we obtain a feasible flow function defined on primal network $G = (N, A, u)$ by $f(i, j) = p(j') - p(i')$ for all $(i, j) \in A$, [7], [8].

An important procedure that will be used in the algorithms from this paper is the shortest paths problem in a planar digraph G. This problem is presented in [13], [14], [15], [16], [17], [18].

We recall the fact that the maximum flow problem into general network with multiple sources and sinks is equivalent with a maximum flow problem into extended general network with a single source and a single sink by introducing an artificial source and an artificial sink and connecting them to all the sources and sinks respectively [1], [4]. In planar network, this extension may destroy the planarity of the network.

Now, we present an algorithm for computing a maximum flow for the case where all the sources and sinks are on the same face (on the outer face). This algorithm is described in the paper [11].

Let $G = (N, A, u)$ be a planar static network and f^* be a maximum flow on G. The first minimum cut $[S^*, \bar{S}^*]$ is defined as follows: S^* is the set of nodes that are reachable from the sources in residual network $\tilde{G}$ with respect to f^*. Let N_4 be $N_4 = N_1 \cup N_3$. The algorithm for maximum flow in directed planar static network with all the sources and sinks are on the outer face (MFDPSNSSOF) is presented below.

Algorithm 2: Algorithm for maximum flow in directed planar static network with all the sources and sinks are on the outer face.

```
MFDPSNSSOF;
BEGIN
     DIV(N_1, N_2, X, Y);
     MFXY(G, X, Y, f_1^*, G~_1);
     MFC(G~_1, C~, f_2^*);
     MFG(f_2^*, G~_2, f_3^*, f_0^*);
END.
```

We now elaborate the procedures of the algorithm. The procedure DIV divides the sources and sinks into two consecutive sets, X and Y, such that $|X_s| = |Y_s|$ and X contains at least as many sources as Y. The procedure MFXY connects all the sources in X to a new node (super source), and all the sinks in Y to a new node (super sink) and computes a maximum flow f_1^* from super source to super sink [4], [7],[9]. Also, this procedure computes the residual network $\tilde{G}_1$ with respect to f_1^*. The procedure MFC deletes the arcs of the first minimum cut $[S_1^*, \bar{S}_1^*]$ from $\tilde{G}_1$ and computes the connected components $\tilde{C} = \{\tilde{C}_1, \ldots, \tilde{C}_i, \ldots, \tilde{C}_k\}$ of $\tilde{G}_1$ and the maximum flow f_{2i}^* in $\tilde{C}_i \in \tilde{C}$ is recursively computed. The residual capacities of the arcs in $\tilde{C}_i$ are the residual capacities in $\tilde{G}_1$. If a connected component C_i contains nodes from both X_s and Y_s, then there can be only two cases: (1) sinks from X with sources and sinks from Y; (2) sources from Y with sinks and sources from X. In the recursive call, in the case (1) we connect all the sinks that belong to X to a super sink, and in the case (2) we connect all the sources that belong to Y to a super source. This is done to ensure that the number of sources and sinks decreases by a constant factor in each recursive call. Also, this procedure computes $f_{2c}^* = f_{21}^* + \cdots + f_{2i}^* + \cdots + f_{2k}^*$ and $f_2^* = f_1^* + f_{2c}^*$. The procedure MFG computes the residual networks $\tilde{G}_2$ with respect to f_2^*, the maximum flow f_3^* from Y to X similar to f_1^*, and the maximum flow $f_0^* = f_2^* + f_3^*$.

From the paper [11] we present following two theorems.

Theorem 4. *The algorithm MFDPSNSSOF correctly computes a maximum flow.*

Theorem 5. *The running time of the algorithm MFDPSNSSOF is* $O(nlog^{1.5}n)$.

The reader which desires more details can consulted the paper [11].

3. Maximum flows in dynamic networks.

Dynamic network models arise in many problem settings, including production distribution systems, economic planning, energy systems, traffic systems, and building evacuation systems.

Let $G = (N, A, u)$ be a static network with the set of nodes $N = \{1, \dots, n\}$, the set of arcs $A = \{a_1, \dots, a_m\}$, the upper bound (capacity) function u, 1 the source node and n the sink node. Let $\mathbb{N}$ be the natural number set and let $H = \{0,1, \dots, T\}$ be the set of periods, where T is a finite time horizon, $T \in \mathbb{N}$. Let use state the transit time function $h: A \times H \to \mathbb{N}$ and the time upper bound function $q: A \times H \to \mathbb{N}$ for all $(i,j) \in A$ and for all $t \in H$. The parameter $h(i,j;t)$ is the transit time needed to traverse an arc (i,j). The parameter $q(i,j;t)$ represents the maximum amount of flow that can travel over arc (i,j) when the flow departs from node i at time t and arrives at node j at time $\theta = t + h(i,j;t)$.

The maximal dynamic flow problem for T time periods is to determine a flow function $g: A \times H \to \mathbb{N}$, which should satisfy the following conditions in dynamic network $D = (N, A, h, q)$:

$$\sum_{t=0}^{T} (g(1, N; t) - \sum_{\tau} g(N, 1; \tau)) = w \tag{4a}$$

$$g(i, N; t) - \sum_{\tau} g(N, i; \tau) = 0, i \neq 1, n, \ t \in H \tag{4b}$$

$$\sum_{t=0}^{T} (g(n, N; t) - \sum_{\tau} g(N, n; \tau)) = -w \tag{4c}$$

$$0 \leq g(i,j;t) \leq q(i,j;t), \quad (i,j) \in A \quad , \quad t \in H \tag{5}$$

$$max \quad w, \tag{6}$$

where $\tau = t - h(k, i; \tau)$, $w = \sum_{t=0}^{T} v(t)$, $v(t)$ is the flow value at time t and $g(i,j;t) = 0$ for all $t \in \{T - h(i,j;t) + 1, \dots, T\}$.

Obviously, the problem of finding a maximum flow in dynamic network $D = (N, A, h, q)$ is more complex than the problem of finding a maximum flow in static network $G = (N, A, u)$. Happily, this complication can be solved by rephrasing the problem in dynamic network D into a problem in static network $R = (V, E, u)$ called the static expanded network[1], [4].

The static expanded network of dynamic network $D = (N, A, h, q)$ is a network $R = (V, E, u)$ with $V = \{i_t | i \in N, t \in H\}$, $E = \{(i_t, j_\theta) | (i,j) \in A, t \in \{0,1, \dots, T - h(i,j;t)\}$, $\theta = t + h(i,j;t), \theta \in H$ }, $u(i_t, j_\theta) = q(i,j;t), (i_t, \ j_\theta) \in E$. The number of nodes in static expanded network R is $n(T+1)$ and number of arcs is limited by $m(T+1) - \sum_A \overset{\circ}{h}(i,j)$, where $\overset{\circ}{h}(i,j) = min\{h(i,j;0), \dots, h(i,j;T)\}$. It is easy to see that any flow in dynamic network D from the source node 1 to the sink node n is equivalent to a flow in static expanded network R from the source nodes $1_0, 1_1, \dots, 1_T$ to the sink nodes $n_0, n_1, \dots, n_T$ and vice versa. We can further reduce the multiple sources, multiple sinks problem in static expanded network R to a single source, single sink problem by introducing a super source node 0 and a super sink node $n+1$ constructing static super expanded network $R_2 = (V_2, E_2, u_2)$, where $V_2 = V \cup \{0, n+1\}$, $E_2 = E \cup \{(0, 1_t) | t \in H\} \cup \{(n_t, n+1) | t \in H\}$, $u_2(i_t, j_\theta) = u(i_t, j_\theta)$, $(i_t, j_\theta) \in E$, $u_2(0, 1_t) = u_2(n_t, n+1) = \infty$, $t \in H$.

Now, we construct the static reduced expanded network $R_1 = (V_1, E_1, u_1)$ as follows. We define the function $h_2: E_2 \longrightarrow \mathbb{N}$, with $h_2(0, 1_t) = h_2(n_t, n+1) = 0$, $t \in H$, $h_2(i_t, j_\theta) = h(i,j;t)$, $(i_t, j_\theta) \in E$. Let $d_2(0, i_t)$ be the length of the shortest path from the source node 0 to the node i_t, and $d_2(i_t, n+1)$ the length of the shortest path from node i_t to the sink node $n+1$, with respect to h_2 in network R_2. The computation of $d_2(0, i_t)$ and $d_2(i_t, n+1)$ for all $i_t \in V$ are performing by means of the usual shortest path algorithms. The network $R_1 = (V_1, E_1, u_1)$ have $V_1 = \{0, n+1\} \cup \{i_t \mid i_t \in V, \ d_2(0, i_t) + d_2(i_t, n+1) \leq T\}$, $E_1 = \{(0, 1_t) \mid d_2(1_t, n+1) \leq T, t \in H\} \cup \{(i_t, j_\theta) \mid (i_t, j_\theta) \in E, \ d_2(0, i_t) + h_2(i_t, j_\theta) + d_2(j_\theta, n+1) \leq T\} \cup \{(n_t, n+1) \mid d_2(0, \ n_t) \leq T, \ t \in H\}$ and u_1 are restrictions of u_2 at E_1.

Now, we construct the static reduced expanded network $R_1 = (V_1, E_1, u_1)$ using the notion of dynamic shortest path. The dynamic shortest path problem is presented in [2]. Let $d(1, i; t)$ be the length of the dynamic shortest path at time t from the source node 1 to the node i and $d(i, n; t)$ the length of the dynamic shortest path at time t from the node i to the sink node n, with respect to h in dynamic network D. Let as consider $H_i = \{t | t \in H, d(1, i; t) \leq t \leq T - d(i, n; t)\}, i \in N$, and $H_{i,j} = \{t | t \in H, d(1, i; t) \leq t \leq T - h(i,j;t) - d(j, n; \theta)\}, (i,j) \in A$. The multiple sources, multiple sinks static reduced expanded network $R_0 = (V_0, E_0, u_0)$ have $V_0 = \{i_t | i \in N, t \in H_i\}$, $E_0 = \{(i_t, j_\theta) \mid (i,j) \in A,\ t \in H_{i,j}\}$, $u_0(i_t, j_\theta) = q(i,j;t), (i_t, j_\theta) \in E_0$. The static reduced expanded network $R_1 = (V_1, E_1, u_1)$ is constructed from network R_0 as follows: $V_1 = V_0 \cup \{0, n+1\}, E_1 = E_0 \cup \{(0, 1_t) | 1_t \in V_0\} \cup \{(n_t, n+1) | n_t \in V_0\}, u_1(0, 1_t) = u_1(n_t, n+1) = \infty,\ 1_t, n_t \in V_0$ and $u_1(i_t, j_\theta) = u_0(i_t, j_\theta),\ (i_t, j_\theta) \in E_0$.

We notice the fact that the static reduced expanded network R_1 is always a partial sub-network of static super expanded network R_2. A dynamic flow for T periods in the dynamic network D is equivalent with a static flow in a static reduced expanded network R_1. Since an item released from a node at a specific time does not return to the location at the same or an earlier time, the static networks R, R_2, R_1, R_0 cannot contain any circuit, and are therefore always acyclic.

In the most general dynamic model, the parameter $h(i) = 1$ is waiting time at node i, and the parameter $q(i;t)$ is upper bound for flow $g(i;t)$ that can wait at node i from time t to $t+1$. This most general dynamic model is not discussed in this paper.

The maximum flow problem for T time periods in dynamic network D formulated in conditions (1), (2) and (3) is equivalent with the maximum flow problem in static reduced expanded network R_0 as follows:

$$\sum_{j_\theta} f_0(i_t, j_\theta) - \sum_{k_\tau} f_0(k_\tau, i_t) = \begin{cases} v_0(i_t), & if\ i_t = 1_t & (7a) \\ 0, & if\ i_t \neq 1_t, n_t & (7b) \\ -v_0(i_t), & if\ i_t = n_t & (7c) \end{cases}$$

$$0 \leq f_0(i_t, j_\theta) \leq u_0(i_t, j_\theta), \quad (i_t, j_\theta) \in E_0 \tag{8}$$

$$\max v_0, \tag{9}$$

where $v_0 = \sum_t v_0(1_t)$.

In the case $h(i,j;t) = h(i,j), t \in H$ the dynamic distances $d(1,i;t), d(i,n;t)$ become static distances $d(1,i), d(i,n)$.

4. Maximum flow in directed planar dynamic networks

In this section we consider that dynamic network $D = (N, A, h, q)$ is $(1,n)$ planar. We construct the multiple sources, multiple sinks static reduced expanded network $R_0 = (V_0, E_0, u_0)$.

Theorem 6. *The multiple sources, multiple sinks static reduced expanded network $R_0 = (V_0, E_0, u_0)$ is a planar static network.*

Proof. Let $P_1, P_2, \dots, P_k$ a directed paths from source node 1 to sink node n in planar dynamic network D with the route time $h(P_i) \leq T$, $i = 1, \dots, k$. The path P_i is repeated for $t = 0,1, \dots, T - h(P_i) + 1$, in network R_0. Because the network D is planar, the arcs of directed paths $P_1, \dots, P_k$ no intersect each other. From the way the reduced extended network R_0 is built result that the arcs of directed paths $P_1^t, P_2^t, \dots, P_k^t$, $t = 0,1, \dots, T - h(P_i) + 1$, $i = 1,2, \dots, k$, no intersect each other, after a new geometric representation of network R_0 (an isomorphic graph with R_0). Therefore R_0 is a planar static network. □

We remark that R_0 is planar static network either with multiple sources and sinks, when all sources and sinks are on the boundary of a single face or with multiple sources and sinks, when the sources and the sinks reside on the boundaries of different faces. Next we consider the first case.

Now we present an algorithm for computing a maximum flow in directed planar dynamic networks. This algorithm is presented below.

Algorithm 3: Algorithm for computing a maximum flow in directed planar dynamic networks

```
MFDPDN;
BEGIN
   CONS(D, R_0);
   MFDPSNSSOF(R_0, f*);
END.
```

The procedure CONS constructs the multiple sources, multiple sinks static reduced expanded network $R_0 = (V_0, E_0, u_0)$ of directed planar dynamic network $D = (N, A, h, q)$. The procedure MFDPSNSSOF is the algorithm presented in Section 2.

Theorem 7. *The algorithm MFDPDN correctly computes a maximum flow in directed planar dynamic network D.*

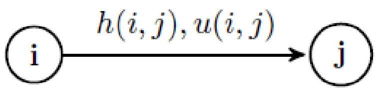
Proof. By Theorem 4 the algorithm MFDPDN correctly computes a maximum flow in directed planar static network R_0. Any flow in network R_0 is equivalent with a flow in network D. Therefore the algorithm MFDPDN correctly computes a maximum flow in directed planar dynamic network D. □

Theorem 8. *The algorithm MFDPDN has the complexity* $O(nTlog^{1.5}nT)$

Proof. By Theorem 5 the MFDPDN has the complexity $O(n_0 log^{1.5} n_0)$. From Section 3 we have $n_0 < nt$. Therefore the algorithm MFDPDN has the complexity $O(nTlog^{1.5}nT)$. □

5. Example

The (1,6) directed planar dynamic network $D = (N, A, h, q)$ is presented in figure 1 and the time horizon set to $T = 6$, therefore $H = \{0,1,2,3,4,5,6\}$. The transit times $h(i,j;t) = h(i,j), t \in H$ and the upper bounds $q(i,j;t) = q(i,j), t \in H$ for all arcs $(i,j) \in A$ are indicated on arcs.



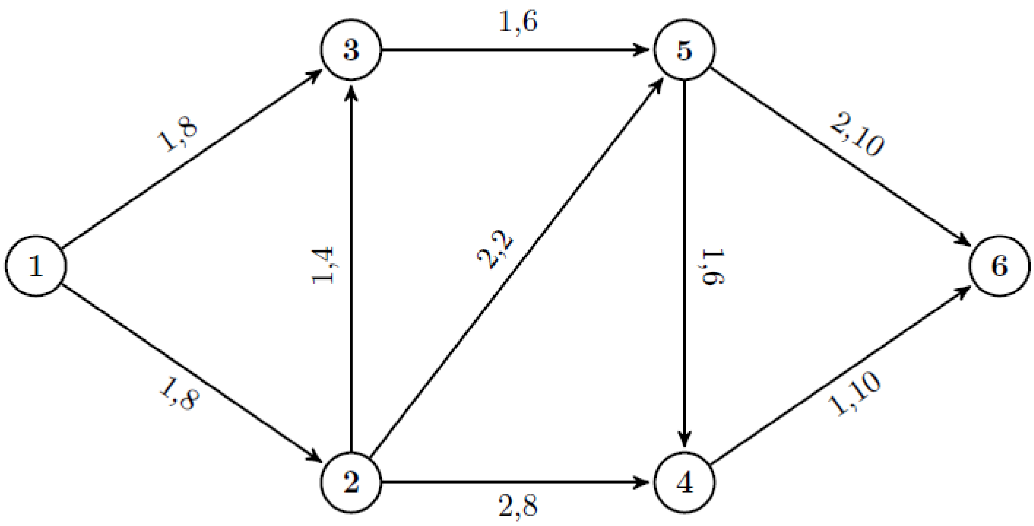


Figure 1: The directed planar dynamic network $D = (N, A, h, q)$

The multiple sources, multiple sinks static reduced expanded network R_0 is presented in figure 2.

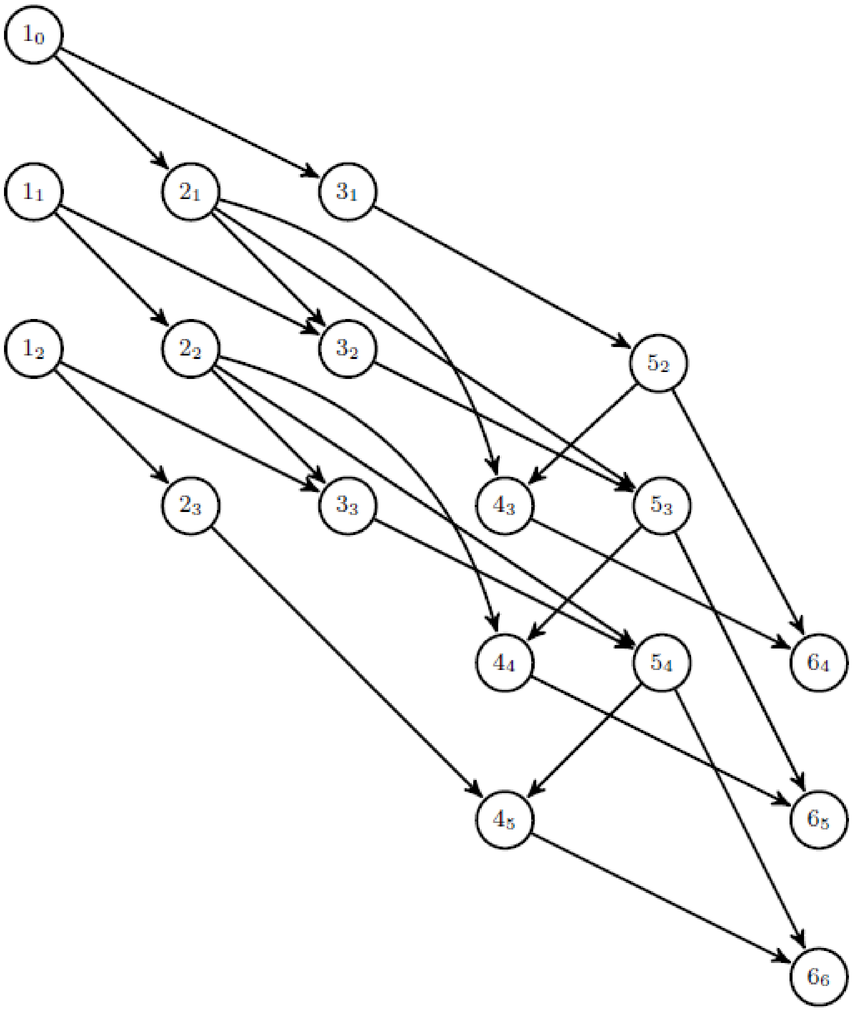


Figure 2: The multiple sources, multiple sinks static reduced expanded network R_0

The directed paths P_i with transit time $h(P_i) \leq T = 6$ and the number of repetitions $n(P_i) = T - h(P_i) + 1$ are presented in Table 1.

Table 1: The directed paths P_i with transit time $h(P_i)$ and the number of repetitions $n(P_i)$

P_i	$h(P_i)$	$n(P_i)$
$P_1 = (1,2,4,6)$	4	3
$P_2 = (1,3,5,6)$	4	3
$P_3 = (1,3,5,4,6)$	4	3
$P_4 = (1,2,5,6)$	5	2
$P_5 = (1,2,3,5,6)$	5	2
$P_6 = (1,2,5,4,6)$	5	2
$P_7 = (1,2,3,5,4,6)$	5	2

A new representation of network R_0, with the algorithm of Demoucron et al., is presented in figure 3.

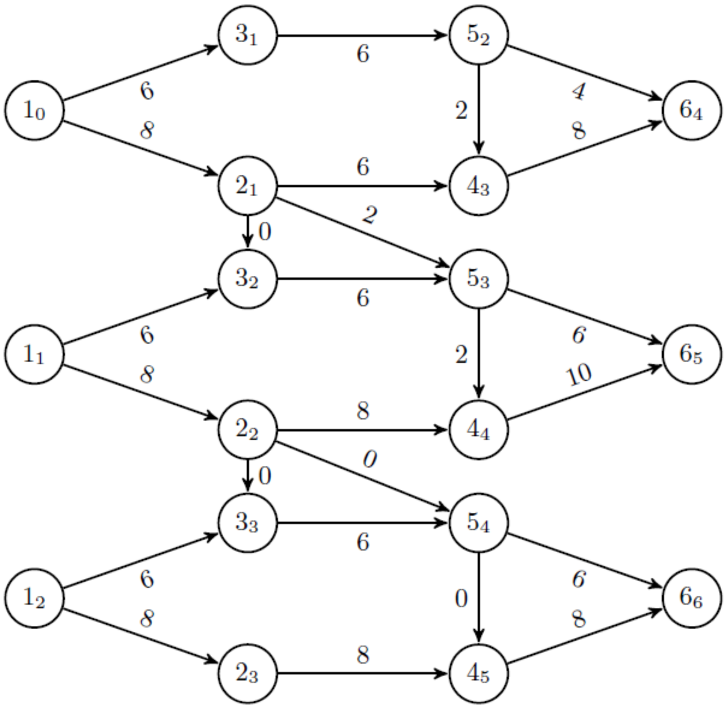


Figure 3: A new representation of network R_0

The procedure DIV from MFDPSNSSOF determines that the source nodes $1_0, 1_1, 1_2$ are in X and the sink nodes $6_4, 6_5, 6_6$ are in Y. The procedure MFXY computes the maximum flow f_1^* which is represented in figure 3 and has the value $v_1^* = 42$. The minimum cut $[S_1^*, \bar{S}_1^*] = (S_1^*, \bar{S}_1^*) \cup (\bar{S}_1^*, S_1^*)$, the set of connected components $\tilde{C}$={$\tilde{C}_1$, $\tilde{C}_2$, $\tilde{C}_3$, $\tilde{C}_4$} and the maximum flow f_2^* are computes with the procedure MFC. We have: $(S_1^*,\bar{S}_1^*)$=$\{(1_0, 2_1), (3_1, 5_2), (1_1, 2_2), (3_2, 5_3), (1_2, 2_3),\ (3_3, 5_4)\} \cup \{(2_1, 3_2), (2_2, 3_2)\}$; $\tilde{C}_1 = \{1_0, 3_1\}$, $\tilde{C}_2 = \{1_1, 3_2\}$, $\tilde{C}_3 = \{1_2, 3_3\}$, $\tilde{C}_4 = \{V_0 - (\tilde{C}_1 \cup \tilde{C}_2 \cup \tilde{C}_3)\}$ (we write only the nodes which generate the connected components). Because $\tilde{C}_1, \tilde{C}_2, \tilde{C}_3$ have not sink nodes and $\tilde{C}_4$ has not source nodes we obtain $f_{21}^* = 0$, $f_{22}^* = 0$, $f_{23}^* = 0$, $f_{24}^* = 0$, therefore $f_{2c}^* = f_{21}^* + f_{22}^* + f_{23}^* + f_{24}^* = 0$ and $f_2^* = f_1^* + f_{2c}^* = f_1^*$. The procedure MFG compute the maximum flow f_3^* and the maximum flow $f_0^* = f_2^* + f_3^*$. Because the set Y has not source nodes (X has not sink nodes) we have $f_3^* = 0$, $f_0^* = f_1^*$ and $v_0^* = v_1^* = 42$. Therefore the minimum cut $[S_1^*, \bar{S}_1^*]$ is a minimum cut in network R_0, $S_0^* = S_1^*$, $\bar{S}_0^* = \bar{S}_1^*$ and $c[S_0^*, \bar{S}_0^*] = c(S_0^*, \bar{S}_0^*) = u(1_0, 2_1) + u(3_1, 5_2) + u(1_1, 2_2) + u(3_2, 5_3) + u(1_2, 2_3) + u(3_3, 5_4) = 8 + 6 + 8 + 6 + 8 + 6 = 42 = v_0^*$.

References

[1] R. AHUJA, T. MAGNANTI and J. ORLIN, *Network Flows. Theory, algorithms and applications*. Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1993.

[2] X. CAI, D. SHA and C. WONG, *Time-varying nework optimization*. Springer, 2007.

[3] G. DEMOUCRON, Y. MALGRANGE and R. PERTUISET *Graphes planaires: reconnaissance et construction de represe′ntations planaires topologiques.* Rev. Francaise Recherche Opérationnélle **8**, 33-47, 1964.

[4] L.R. FORD and D.R. FULKERSON, *Maximal flows throgh a network*. Canadian Journal Mathematics **8**, 399-404, 1956.

[5] G. BORRADAILE and P.N. KLEIN, *An $O(nlogn)$ algorithm for maximum st-flow in a directed planar graph* Journal ACM , no.2, 2009.

[6] G. BORRADAILE, P.N. KLEIN, S. MOZES, Y. NUSSBAUM, and C. WULF-NILSEN *Multiple-source multiple-sink maximum flow in directed planar graph in near-lineare time*. In *Proceedings of the 52nd Annual Symposium on Foundations of Computer Science* 170-179, 2011.

[7] R. HASSIN, *Maximum flow in (s,t) planar networks*. Information Processing Letters **13**, no.3, 107, 1981.

[8] R. HASSIN and D.B. JOHNSON, *An $O(nlog^2n)$ algorithm for maximum flow in undirected planar networks*. SIAM Journal on Computing **14**, 612-624, 1985.

[9] A. ITAI and Y. SHILOACH, *Maximum flow in planar networks*. SIAM Journal on Computing **8**, 135-150, 1979.

[10] D.B. JOHNSON and S. VENKATESAN, *Using divide and conquer to find flows in directed planar networks in $O(n^{3/2}logn)$ time.* In *Proceedings of the 20th Annual Allerton Conference on Comunication, Control, and Computing* 898-905, 1982.

[11] G.L. MILLER and J. NAOR, *Flow in planar graphs with multiple sources and sinks.* SIAM Journal on Computing **24**, no.5, 1002-1017, 1995.

[12] C. SCHIOPU and E. CIUREA, *The maximum flows in planar dynamic networks. International* Journal of Computers Communications and Control **11**, no.2, 282–291, 2016.
[13] J. FAKCHAROENPHOL and S. RAO, *Planar graphs, negative weight edges, shortest paths, and near linear time.* Journal of Computer and System Sciences **72**, no.5, 868-889, 2006.
[14] G. FREDERIKSON, *Fast algorithms for shortest paths in planar graph with applications.* SIAM Journal of Computing **16**, 1004-1022, 1987.
[15] M.R. HENZINGER, P.N. KLEIN, S. RAO and S. SUBRAMANIAN, *Faster shortest-path algorithms for planar graphs.* Journal of Computer and System Sciences **55**, no.1, 3-23, 1997.
[16] P.N. KLEIN, *Multiple-source shortest path in planar graphs.* In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithm* 146-155, 2005.
[17] P.N. KLEIN and S. SUBRAMANIAN, *A fully dynamic approximation scheme for shortest paths in planar graph.* Algorithmica **22**, no.3, 235-249, 1998.
[18] S. MOZES and C. WULFF-NILSEN, *Shortest path in planar graphs with real lengths in* $O(nlog^2n/loglogn)$ *time. Proceedings of the 18th European Symposium on Algorithms* 206–217, 2010.