

The Maximum Flows in Planar Dynamic Networks

C. Schiopu, E. Ciurea

Camelia Schiopu*

1. Transilvania University of Brasov
Romania, 500091 Braşov, Iuliu Maniu, 50
*Corresponding author: camelia.s@unitbv.ro

Eleonor Ciurea

Transilvania University of Brasov
Romania, 500091 Braşov, Iuliu Maniu, 50
e.ciurea@unitbv.ro

Abstract: An nontrivial extension of the maximal static flow problem is the maximal dynamic flow model, where the transit time to traverse an arc is taken into consideration. If the network parameters as capacities, arc traversal times, and so on, are constant over time, then a dynamic flow problem is said to be stationary. Research on flow in planar static network is motivated by the fact that more efficient algorithms can be developed by exploiting the planar structure of the graph. This article states and solves the maximum flow in directed $(1, n)$ planar dynamic networks in the stationary case.

Keywords: network flow, planar network, dynamic network, maximum flow.

1 Introduction

The static network flow models bridges several diverse and seemingly unrelated areas of combinatorial optimization. More often in scientific writing, flow in a network refers to the flow of electricity, phone calls, email messages, commodities being transported across truck routes, or other such kinds of flow. Many efficient algorithms have been developed to solve the maximum flows problem in static network [1].

The planar static network also arise in practical contexts such as VLSI design and communication networks, and hence it is of interest to find fast flow algorithms for this class of graphs. The computation of a maximum flow in a planar static network has been investigated by many researchers starting from the work of Ford and Fulkerson [5] who developed an $O(n^2)$ time algorithm for $(1, n)$ networks when the source node 1 and sink node n are on the same face. This algorithm was later improved to $O(n \log n)$ time by Itai and Shiloach [8]. By introducing the concept of potentials, Hassin [6] gave an algorithm that run in $O(n \log^{0.5} n)$ time using Frederickson's shortest path algorithm [4]. Itai and Shiloach [8] also developed an algorithm to find a maximum flow in an undirected planar networks when the source node and sink node are not on the same face. For faster maximum flow algorithms in planar (but not necessarily $(1, n)$ planar) undirected and directed static networks see Hassin and Johnson [7] and Johnson and Venkatesan [9]. Khuller and Naor [10] present the flow in planar static networks with nodes capacities.

However, in some other applications, time is an essential ingredient [1]. In this instance, to account properly for the evolution of the underlying system over time, we need to use dynamic network flow models. For dynamic network flow problem see [1], [2], [3].

In this paper, we present the maximum flow problem in directed $(1, n)$ planar dynamic networks. We present the case when the planar dynamic network is stationary. Further on, in Section 2 the maximum flow in directed $(1, n)$ planar static network is exposed. In Section 3 some

basic dynamic network notations and results are presented, while in Section 4 is presented the method for solving the maximum flow in directed $(1, n)$ planar dynamic network. The conclusions are presented in Section 5 and an example is given in Section 6.

2 The maximum flow in directed $(1, n)$ planar static network

Research on flow in planar static network is motivated by the fact that more efficient algorithms can be developed by exploiting the planar structure of the digraph.

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.

Researchers have developed very efficient algorithms (in fact, linear time algorithms) for testing the planarity of a digraph.

Definition 2. Let $G = (N, A)$ be a planar digraph. A face of 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 face x is the set of all arcs that enclose it. Faces x and y said to be adjacent if their boundaries contain a common arc.

The planar digraph G has an unbounded face.

Recall two well-known properties of planar digraphs:

- If a connected planar digraph has n nodes, m arcs and q faces, then $q = m - n + 2$;
- If a planar digraph has n nodes and m arcs, then $m < 3n$.

Our discussion in this paper applies to a special class of planar digraphs known as $(1, n)$ planar digraphs (the node source 1 and node sink n lie on the boundary of unbounded face).

Let $G = (N, A, u)$ be a 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 the upper bound (capacity) function $u : A \rightarrow \mathbb{R}_+$, where \mathbb{R} is real number set. To define maximal static flow problem, we distinguish two special nodes in the static network $G = (N, A, u)$: a source node 1 and a sink node n .

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

$$\sum_j f(i, j) - \sum_k f(k, i) = \begin{cases} v, & \text{if } i = 1 \\ 0, & \text{if } i \neq 1, n \\ -v, & \text{if } i = n \end{cases} \quad \begin{matrix} (1a) \\ (1b) \\ (1c) \end{matrix}$$

$$0 \leq f(i, j) \leq u(i, j), \quad (i, j) \in A \quad (2)$$

for some $v \geq 0$.

We refer to v as the value of the static flow f . The maximum flow problem is to determine a flow f which maximizes v . A cut is a partition of the node set N into two subsets 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 (j, i) with $j \in \bar{S}$ and $i \in S$ 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 $1 - n$ cut if $1 \in S$ and $n \in \bar{S}$.

For the maximum flow problem, we define the capacity of the $1 - n$ cut $[S, \bar{S}]$ as:

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

We refer to an $1 - n$ cut whose capacity is the minimum among all $1 - n$ cuts as a minimal cut.

Recall the maximum flow minimum cut theorem.

Theorem 3. *The maximum value of the flow from a source node 1 to a sink node n in network G equals the capacity of minimum $1 - n$ cut.*

Many efficient algorithms have been developed to solve the maximum flows problem in some static network [1].

Next we present the maximum flow problem in directed $(1, n)$ planar static network. First, we define the dual directed static network denoted by $G' = (N', A', c')$. We add the arc $(n, 1)$ with $u(n, 1) = 0$, which divides the unbounded face into two faces: a new bounded face and a new unbounded face. In this case we have $n' = q + 1$ faces, with $q = m - n + 2$. Then we place a node x' inside each face x of the network G . We have $N' = \{1', \dots, x', \dots, y', \dots, n'\}$. Let $1'$ and n' , respectively, denote the nodes in the dual directed static network G' corresponding to the new bounded face and the new unbounded face. Each arc $(i, j) \in A$ lies on the boundary of the two faces x and y . Corresponding to this arc, the network G' contains two oppositely arcs (x', y') and (y', x') . If arc (i, j) is a clockwise arc in the face x , we define the cost $c'(x', y') = u(i, j)$ and the cost $c'(y', x') = 0$. We define arc costs in the opposite manner if arc (i, j) is a counterclockwise arc in the face x . The network G' contains the arcs $(1', n')$ and $(n', 1')$ which we delete from the network. We have $A' = \{(x', y'), (y', x') | x', y' \in N', (x', y') \text{ and } (y', x') \text{ correspond to } (i, j) \in A\}$. There is an one-to-one correspondence between $1 - n$ cuts in the network G and paths from node $1'$ to node n' in the network G' . Moreover, the capacity of the cut equals the cost of the corresponding path. Consequently, we can obtain a minimum $1 - n$ cut $[S, \bar{S}]$ and $c[S, \bar{S}]$ in network G by determining a shortest path P' and $c'(P')$ from node $1'$ to node n' in the network G' . We can solve the shortest path problem in the network G' using Dijkstra's algorithm [1].

Now, we present an algorithm for finding a maximum flow in a directed $(1, n)$ planar static network $G = (N, A, u)$. Let $d'(x')$ denote the shortest path distance from node $1'$ to node x' in the dual directed static network $G' = (N', A', c')$. The algorithm for maximum flow in directed $(1, n)$ planar static network (AMFDPSN) is presented in Figure 1 [1].

- 1: AMFDPSN;
- 2: begin
- 3: compute the network G' ;
- 4: DIJKSTRA (G', d');
- 5: **for** $(i, j) \in A$ **do**
- 6: $f(i, j) := d'(y') - d'(x')$;
- 7: **end for**
- 8: end.

Figure 1: Algorithm for maximum flow in directed $(1, n)$ planar static network

Theorem 4. *The AMFDPSN determines a maximum flow in network G .*

Theorem 5. *The AMFDPSN determines a maximum flow in $O(n^2)$ time.*

Using Frederickson's algorithm (see [4]), the AMFDPSN determines a maximum flow in $O(n^{1.5})$ time.

3 The maximum flows in dynamic network

Let $G = (N, A, u)$ be a static network with the node set N , the arc set A , the upper bound 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 us state the transit time function $h : A \times H \rightarrow \mathbb{N}$ and the time capacity function $u_h : A \times H \rightarrow \mathbb{R}_+$, where $h(i, j; t)$ represents the transit time of arc (i, j) at time t , $t \in H$ and $u_h(i, j; t)$ represents the capacity (upper bound) of arc (i, j) at time t , $t \in H$.

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

$$\sum_{t=0}^T (\sum_j f_h(i, j; t) - \sum_k \sum_{\tau} f_h(k, i; \tau)) = v_H, \text{ if } i = 1 \tag{4a}$$

$$\sum_j f_h(i, j; t) - \sum_k \sum_{\tau} f_h(k, i; \tau) = 0, \text{ if } i \neq 1, n, \quad t \in H \tag{4b}$$

$$\sum_{t=0}^T (\sum_j f_h(i, j; t) - \sum_k \sum_{\tau} f_h(k, i; \tau)) = -v_H, \text{ if } i = n \tag{4c}$$

$$0 \leq f_h(i, j; t) \leq u_h(i, j; t), \text{ for all } (i, j) \in A \text{ and for all } t \in H \tag{5}$$

$$\max v_H, \tag{6}$$

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

In other words, a dynamic flow f_h from 1 to n is any flow f_h from 1 to n in which not more than $u_h(i, j; t)$ flow units starting from node i at time t and arriving at node j at time $t + h(i, j; t)$ for all arcs (i, j) and all t . Note that in a dynamic flow, units may be departing from the source at time $0, 1, \dots, T'$, $T' < T$. A maximum dynamic flow for T time periods from 1 to n is any dynamic flow from 1 to n in which the maximum possible number of flow units arrive at the sink node n during the first T time periods. We will show how to transform the maximum dynamic flow problem in the dynamic network $G_h = (N, A, h, u_h)$ into a static flow problem on a static network $G'_H = (N'_H, A'_H, u'_H)$, called the reduced time-expanded network.

For a given dynamic network $G_h = (N, A, h, u_h)$, we form the time expanded network $G_H = (N_H, A_H, u_H)$ as follows. We make $T + 1$ copies i_t , $t = 0, 1, \dots, T$ of each node i in G_h . Node i_t in G_H represents node i in G_h at time t . For each (i, j) in G_h , there are arcs (i_t, j_θ) , $\theta = t + h(i, j; t)$, $t = 0, 1, \dots, T - h(i, j; t)$ with capacity $u_H(i_t, j_\theta) = u_h(i, j; t)$ in G_H . The arc (i_t, j_θ) in G_H represents the potential movement of a commodity from node i to node j in time $h(i, j; t)$. The number of nodes in G_H is $n(T + 1)$, and number of arcs is limited by $m(T + 1) - \sum_A \bar{h}(i, j)$, where $\bar{h}(i, j) = \min\{h(i, j; 0), \dots, h(i, j; T)\}$. It is easy to see that any dynamic flow in dynamic network G_h is equivalent to a static flow in static network G_H 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 source, multiple sink problem in network G_H to the single source, single sink problem by introducing a supersource node 1^* and a supersink node n^* constructing time superexpanded network $G_H^* = (N_H^*, A_H^*, u_H^*)$, where $N_H^* = N_H \cup \{1^*, n^*\}$, $A_H^* = A_H \cup \{(1^*, 1_t) | t = 0, 1, \dots, T\} \cup \{(n_t, n^*) | t = 0, 1, \dots, T\}$, $u_H^*(i_t, j_\theta) = u_H(i_t, j_\theta)$ for all $(i_t, j_\theta) \in A_H$, $u_H^*(1^*, 1_t) = u_H^*(n_t, n^*) = \infty$, $t = 0, 1, \dots, T$. Now, we construct the time reduced

expanded network $G'_H = (N'_H, A'_H, u'_H)$ as follows. We define the function h^* , $h^* : A^*_H \rightarrow \mathbb{N}$, $h^*(1^*, 1_t) = h^*(n_t, n^*) = 0$, $t = 0, 1, \dots, T$, $h^*(i_t, j_\theta) = h(i, j; t)$, $t = 0, 1, \dots, T - h(i, j; t)$. Let $d^*(1^*, i_t)$ be the length of the shortest path from the source node 1^* to the node i_t in network G^*_H and $d^*(i_t, n^*)$ the length of the shortest path from node i_t to the sink node n^* , with respect to h^* . The computation of $d^*(1^*, i_t)$ and $d^*(i_t, n^*)$ for all $i_t \in N^*_H$ is performed by means of the usual shortest path algorithms. We have $N'_H = \{1^*, n^*\} \cup \{i_t | i_t \in N_H, d^*(1^*, i_t) + d^*(i_t, n^*) \leq T\}$, $A'_H = \{(1^*, 1_t) | d^*(1_t, n^*) \leq T\} \cup \{(n_t, n^*) | d^*(1^*, n_t) \leq T\} \cup \{(i_t, j_\theta) | (i_t, j_\theta) \in A_H, d^*(1^*, i_t) + h^*(i_t, j_\theta) + d^*(j_\theta, n^*) \leq T\}$ and u'_H is restriction of u^*_H at A'_H . In network G'_H we rewrite the nodes $1^*, n^*$ by $1',$ respectively n' . It is easy to see that the network G'_H is always a partial subnetwork of G^*_H . Since an item released from a node at a specific time does not return to that location at the same or an earlier time, the networks G_H, G^*_H, G'_H cannot contain any circuit, and are therefore acyclic always.

In the most general dynamic model, the parameter $h(i) = 1$ is waiting time at node i , and the parameter $u_h(i; t)$ is defined as the capacity of the node i , which represents the maximum amount of flow 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 dynamic flow problem for T time periods in dynamic network G_h formulated in conditions (4), (5), (6) is equivalent with the maximum static flow problem in static network G'_H as follows:

$$\sum_{j_\theta} f'_H(i_t, j_\theta) - \sum_{k_{\tau'}} f'_H(k_{\tau'}, i_t) = \begin{cases} v'_H, & \text{if } i_t = 1', \\ 0, & \text{for all } i_t \neq 1', n', \\ -v'_H, & \text{if } i_t = n', \end{cases} \tag{7a}$$

$$0 \leq f'_H(i_t, j_\theta) \leq u'_H(i_t, j_\theta), \text{ for all } (i_t, j_\theta) \in A'_H \tag{8}$$

$$\max v'_H, \tag{9}$$

where by convention $i_t = 1'$ for $t = -1$ and $i_t = n'$ for $t = T + 1$.

It is easy to see that network G'_H is no planar, in general.

A dynamic flow problem is said to be stationary if the network parameters as capacities, arc traversal times, and so on, are constant over time ($c : A \rightarrow \mathbb{R}_+, h : A \rightarrow \mathbb{N}$, and so on). In the stationary case it does not require the construction of the reduce time expanded static network $G'_H = (N'_H, A'_H, u'_H)$ for solving the maximum dynamic flow problem for any T . A maximum dynamic flow in the stationary case can be generated from a maximum value and minimum time flow f in static network $G = (N, A, c, u)$, where $c(i, j) = h(i, j)$ is the cost for any arc $(i, j) \in A$. The algorithm for stationary maximum dynamic flow (ASMDF) problem is presented in Figure 2 [5].

- 1: ASMDF;
- 2: BEGIN
- 3: AMVMCF (G,f);
- 4: ADFEF ($f, r(P_1), \dots, r(P_k)$);
- 5: ARF ($r(P_1), \dots, r(P_k)$);
- 6: END.

Figure 2: Algorithm for stationary maximum dynamic flow.

The procedure AMVMCF performs the algorithm for maximum value and minimum cost flow f in network G . For statements we suppose that use the algorithm of Klein variant (minimum

mean cycle canceling algorithm, see [1]). This algorithm have the complexity $O(n^2m^3 \log n)$. The procedure ADFFEF performs the algorithm for decomposition of flow f in elementary flows with $r(P_1), \dots, r(P_k)$ path flows. Is necessary that $c(P_i) \leq T$. This algorithm have complexity $O(m^2)$. The procedure ARF performs the algorithm for send $r(P_i)$ flow, $i = 1 \dots, k$, starting out from source node 1 at time periods 0 and repeat it after each time period as long as there is enough time left in the horizon for the flow along the path to arrive at the sink node n . This algorithm have complexity $O(kT)$. Hence, the algorithm for stationary maximum dynamic flow have complexity $O(n^2m^3 \log n)$ (we consider that $kT \leq n^2m^3 \log n$). The flow obtained with ASMDF is called a temporally repeated flow for the obvious reason that is consists of repeated shipments along the same flow paths from 1 to n . The maximum value of a temporally repeated flow obtained with ASMDF is:

$$v_H = (T + 1)v - \sum_A h(i, j)f(i, j) \quad (10)$$

where v is the maximum value of the flow f obtained with AMVMCF.

4 The maximum flows in planar dynamic networks

In this section we consider the maximum flows in planar dynamic networks in the stationary case. Hence, we use the ASMDF which has presented in Section 3. The network $G = (N, A, c, u)$ is planar.

The minimum mean cycle canceling algorithm is a special case of the Klein's algorithm (cycle canceling algorithm, see [1]). Recall that the mean cost of a directed cycle (circuit) \dot{P} is $(\sum(c(i, j)|(i, j) \in \dot{P}))/|\dot{P}|$, and that the minimum mean cycle is a cycle with the smallest mean cost in the network G . Is known that use dynamic programming algorithm to find the minimum mean cycle in $O(nm)$ time, see [1].

In this case, the minimum mean cycle canceling algorithm starts with a maximum flow f in the network G . This flow is computed with algorithm presented in Section 2. At every iteration, the minimum mean cycle canceling algorithm identifies a minimum mean cycle \dot{P} in residual network \tilde{G} . If the mean cost of the cycle \dot{P} is negative, the algorithm augments the maximum possible flow along \dot{P} , updates \tilde{G} , and repeats this process. If the mean cost of \dot{P} is nonnegative, \tilde{G} contains no negative cycle and f is a maximum value and minimum cost flow, so the algorithm terminates. This algorithm is surprisingly simple to state.

Theorem 6. *The ASMDS correctly computes the maximum flow in planar stationary dynamic network.*

Proof: The ASMDS correctly computes the maximum flow in general stationary dynamic network. Obviously that algorithm is correctly and for planar network. \square

Theorem 7. *The ASDMS applied in planar network has the complexity $O(n^5 \log n)$.*

Proof: The ASMDF applied in general network has the complexity $O(n^2m^3 \log n)$. In planar network we have $m = O(n)$. Hence, the ASMDF applied in planar network has the complexity $O(n^5 \log n)$. \square

5 Conclusions

The computation of a maximum flow in a general network has been an important and well studied problem, both in the fields of computer science and operations research. Many efficient

algorithms have been developed to solve this problem, see, e.g., [1]. Research on maximum flow in planar network is motivated by the fact that more efficient algorithms can be developed by exploiting the planar structure of the graph. The planar flow algorithms are not only extremely efficient, but they are also very elegant. Planar networks also arise in practical contexts such as VLSI design and communication networks, and hence it is of interest to find fast flow algorithms for this class of networks.

In this paper, we have studied a generalization of the maximum flow in directed $(1, n)$ planar networks, to include transit time features encountered in many practical situations. The our model, assumes that all attributes in the problem, including arc capacities and transit times, do not change over time. In this case we have used an efficient procedure to find the maximum value and minimum cost flow in directed $(1, n)$ planar static networks $G = (N, A, c = h, u)$, and then develops a set of temporally repeated flows, with the optimal flow decomposed into a set of path flows. We remark that the problem of maximum flow in $(1, n)$ planar dynamic networks was not studied up to the present. Also, we introduce the notion of reduced time expanded network $G'_H = (N'_H, A'_H, u'_H)$ and show how make this network.

Future research directions include problems:

- (1) the maximum flow in directed $(1, n)$ planar dynamic networks, where the transit times, the capacities of arcs are all time-varying;
 - (2) the maximum flow in directed $(1, n)$ planar dynamic networks with lower bounds in stationary case and in nonstationary case.
- These are more practical features in many real-world problems where we desire to control the speed of flows at different arcs.

6 Example

The planar dynamic network is presented in Figure 3(a) and time horizon being set to $T = 4$, therefore $H = \{0, 1, 2, 3, 4\}$. The transit times $h(i, j)$ and the upper bounds (capacities) $u(i, j)$ for all arcs are indicate in Figure 3(b).

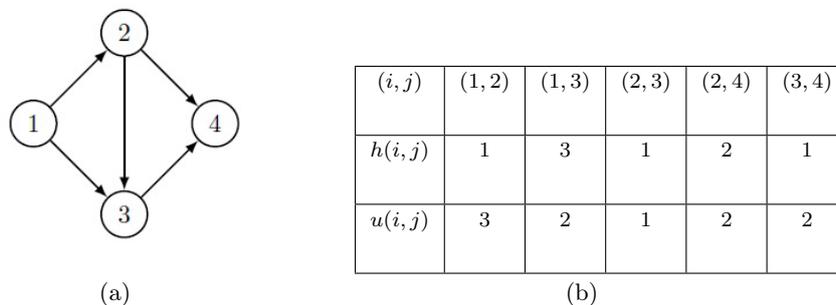


Figure 3: The planar dynamic network

Figure 4 shows the $(1', 4')$ dual network $G' = (N', A', c')$ corresponding to network G_h .

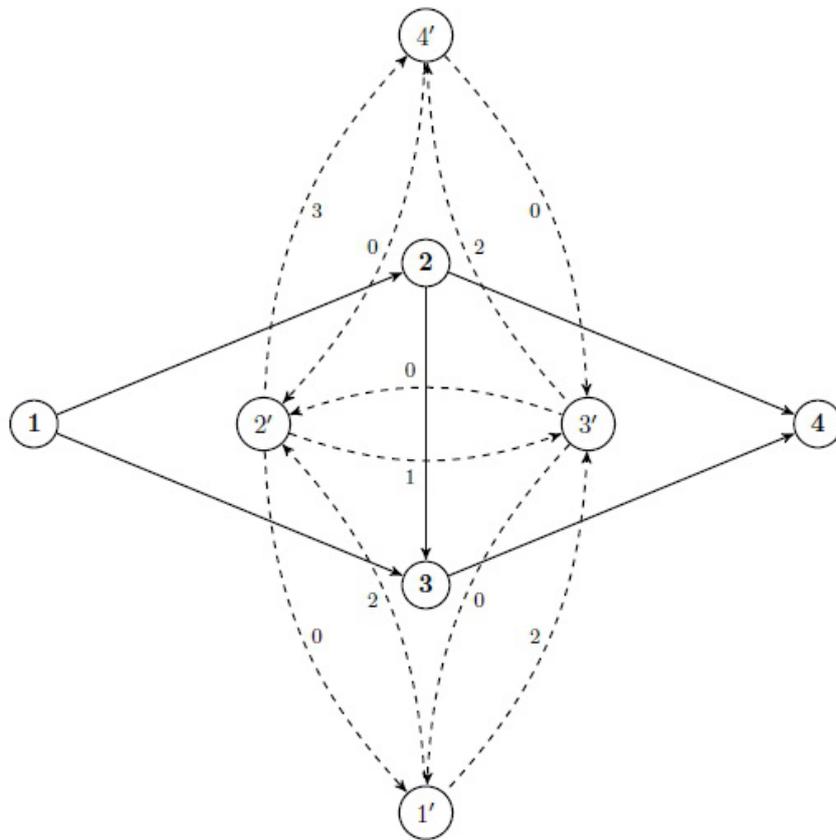


Figure 4: Dual network G' corresponding to the network G

The flow obtained with AMFDPSN is presented in Figure 5(a).

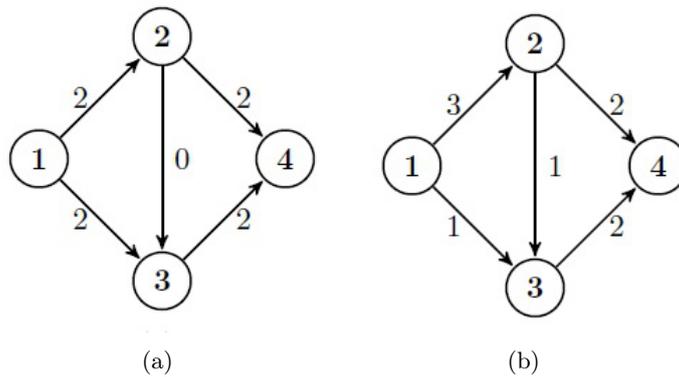


Figure 5: (a) maximum flow; (b) maximum flow of minimum cost

With the minimum mean cycle canceling algorithm we obtain the maximum flow of minimum cost and is presented in Figure 5(b). Applying the procedure ADFEF we have the following path: $P_1 = (1, 2, 4), h(P_1) = 3, r(P_1) = 2$; $P_2 = (1, 2, 3, 4), h(P_2) = 3, r(P_2) = 1$; $P_3 = (1, 3, 4), h(P_3) = 4, r(P_3) = 1$. With the procedure ARF we obtain the maximum dynamic flow which is shown in network $G'_H = (N'_H, A'_H, u'_H)$ in Figure 6.

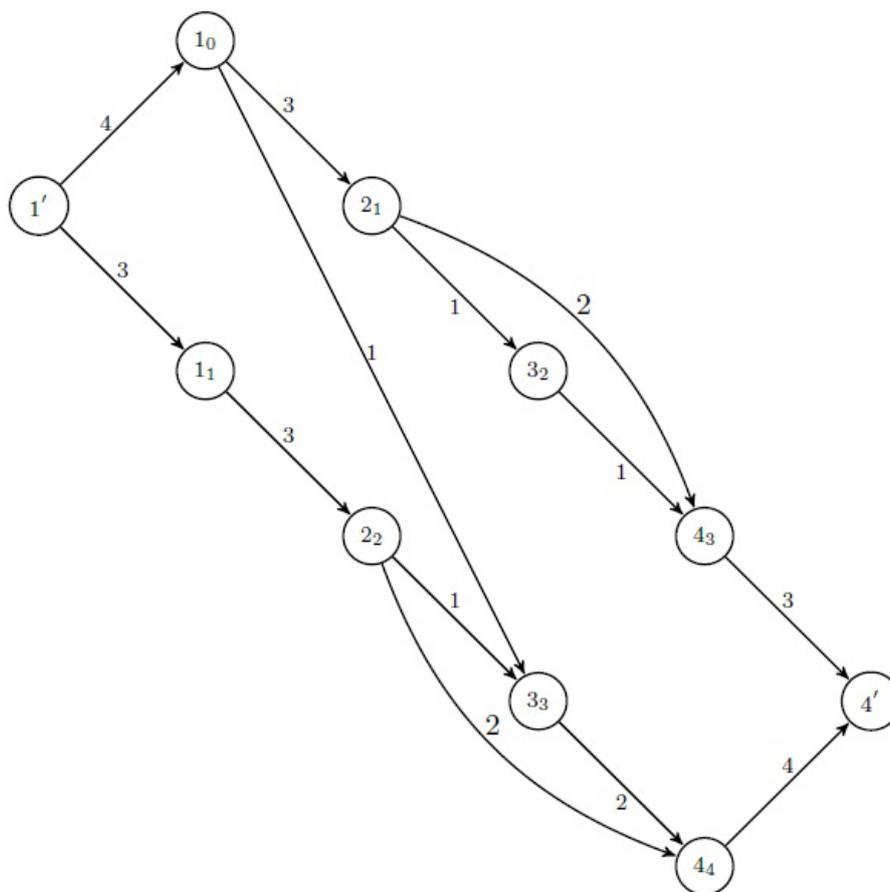


Figure 6: The maximum dynamic flow

Applying the formula (10) we have $v_H = (4 + 1)4 - (3 + 3 + 1 + 4 + 2) = 7$.

For $S'_H = \{1', 1_0, 1_1, 2_2, 3_3\}$, $\bar{S}'_H = \{2_1, 3_2, 4_3, 4_4, 4'\}$ we have $[S'_H, \bar{S}'_H] = (S'_H, \bar{S}'_H) = \{(1_0, 2_1), (2_2, 4_4), (3_3, 4_4)\}$ and $v'_H = v_H = f'_H(S'_H, \bar{S}'_H) = u'_H(S'_H, \bar{S}'_H) = 3 + 2 + 2 = 7$.

Bibliography

- [1] Ahuja,R.; Magnanti,T.; Orlin,J. (1993); *Network Flows. Theory, algorithms and applications*, Editing Prentice hall, Inc.,Englewood Clifss, New Jersey.
- [2] Cai,X.; Sha,D.; Wong,C. (2007); *Time-varying Network Optimization*, Editing Springer.
- [3] Ciurea,E. (1984); Les problemes des flots dynamiques, *Cahiers du CERO*, 26(1-2): 3-9.
- [4] Frederickson,G. (1987); Fast algorithms for shortest path in planar graphs, with applications, *SIAM Journal on Computing*, 16: 1004-1022.
- [5] Ford,L.; Fulkerson,D. (1962); *Flows in Networks*, Princeton University Press, Princeton, N.J.
- [6] Hassin,R. (1981); Maximum flows in (s, t) planar networks, *Information Processing letters*, 13: 107.

-
- [7] Hassin,R.; Johnson,D. (1985); An $O(n \log^2 n)$ algorithm for maximum flow in undirected planar networks, *SIAM Journal on Computing*, 14: 612-624.
- [8] Itai, A.; Shiloach, Y. (1979); *Maximum flow in planar networks*, SIAM Journal on Computing, 8: 135-150.
- [9] Johnson, D.; Venkatesan,S. (1982); Using divide and conquer to find flows in directed planar networks in $O(n^{1.5} \log n)$ time, *Proceedings of the 20th Annual Allerton Conference on Communication , Control and Computing*, University of Illinois, Urbana-Champaign, IL., 898-905.
- [10] Khuller,S.; Naor,J. (1994); Flows in planar graphs with vertex capacities, *Algorithmica*, 11: 200-225.