An Efficient Algorithm for the Evacuation Problem in a Certain Class of a Network with Uniform Path-Lengths

Abstract. In this paper, we consider the evacuation problem for a network which consists of a directed graph with capacities and transit times on its arcs. This problem can be solved by the algorithm of Hoppe and Tardos [1] in polynomial time. However their running time is high-order polynomial, and hence is not practical in general. Thus it is necessary to devise a faster algorithm for a tractable and practically useful subclass of this problem. In this paper, we consider a dynamic network with a single sink s such that (i) for each vertex v the sum of transit times of arcs on any path from v to s takes the same value, and (ii) for each vertex v the minimum v-s cut is determined by the arcs incident to s whose tails are reachable from v. We propose an efficient algorithm for this network problem. This class of networks is a generalization of the grid network studied in the paper [2].

1 Introduction

The problem for finding the most effective plan to evacuate people to safe place has been modelled as an *evacuation problem* by using *dynamic network flow*. In the evacuation problem, we are given a directed graph $D = (V, A)$ which consists of a vertex set V of n vertices with supply $b(v)$ on every vertex v and an arc set A of m arcs with capacity $c(e)$ and transit time $\tau(e)$ on every arc e and a single sink $s \in V$. If we consider urban evacuation, vertices model buildings, rooms, exits and so on, and arcs model pathways or roads. For an arc e, capacity $c(e)$ represents the number of people which can traverse e per unit time, and transit time $\tau(e)$ represents the time required to traverse e. For any vertex v, supply $b(v)$ represents the number of people which exist at v. The evacuation problem asks to find the minimum time required to send all the supplies to a sink.

Given time horizon T, the decision problem of whether we can send all supplies to a sink within time horizon T can be transformed to the maximum-flow problem defined on the *time-expanded network* introduced by Ford and Fulkerson [3]. However the time-expanded network consists of $O(T)$ copies of original vertices and arcs and hence does not lead to the efficient algorithm.

The first polynomial time algorithm for the evacuation problem was proposed by Hoppe and Tardos [1]. However it requires to use the submodular function minimization as a subroutine. Hence the running time is high-order polynomial, and the algorithm is not practical in general. Therefore it is necessary to devise a faster algorithm for a tractable and practically useful subclass of this problem.

As a special case, Mamada et al. [4] gave $O(n \log^2 n)$ time algorithm for the tree network. Hall et al. [5] studied the case called *uniform path-lengths* where there exists a single sink s and for any vertex v the sum of transit times of arcs on any path from v to s takes the same value. They showed that in this case the time-expanded network can be condensed to the so-called *condensed time-expanded network* whose size is polynomial in the input size. Kamiyama et al. [2] have shown an $O(n \log n)$ time algorithm for a $\sqrt{n} \times \sqrt{n}$ grid network with uniform arc capacity.

In this paper, we will generalize the class of networks for which the ideas developed in [2] can be applied, i.e., we consider a dynamic network with a single sink s such that (i) for each vertex v the sum of transit times of arcs on any path from v to s takes the same value, and (ii) for each vertex v the minimum v-s cut is determined by the arcs incident to s whose tails are reachable from v. The algorithm of [2] reduced the evacuation problem to the *min-max resource allocation problem* [6], but in this paper we reduce the evacuation problem to the *parametric flow problem* defined on a static network[1]. Although it is known [5] that the evacuation problem in the case of uniform path-lengths can be reduced to the parametric flow problem in which the capacity of a subset of arcs is a linear function of time horizon T, we prove that in our case the evacuation problem can be reduced to the special case of the parametric flow problem studied by [7] which can be solved more efficiently than the general parametric flow problem considered in [5]. Thus in the case where the input dynamic network satisfies (i) and (ii), our algorithm is faster than using the condensed time-expanded network. In particular, our algorithm becomes much faster when the in-degree of a sink is small or considered to be a constant which is often the case with road networks.

2 Preliminaries

Let $\mathbb{R}_+$ and $\mathbb{Z}_+$ denote the set of nonnegative reals and nonnegative integers, respectively. We will not distinguish between a singleton $\{x\}$ and its element x. For any finite set X, we define $|X|$ as the number of elements that belong to X.

Directed graph. We denote by $D = (V, A)$ a directed graph which consists of a vertex set V and an arc set A. A vertex u is said to be *reachable* to a vertex v when there is a path from u to v. We denote by $e = uv$ an arc e whose tail is u and head is v. For any $X, Y \subseteq V$, we define $\delta(X, Y) = \{e = xy \colon x \in X, y \in Y\}$, and we write $\delta^+(X)$ and $\delta^-(X)$ instead of $\delta(X, V - X)$ and $\delta(V - X, X)$, respectively. For any $u, v \in V$, we denote by $\lambda_D(u, v)$ the local arc connectivity from u to v in D. For any $W \subseteq V$, let $D[W]$ denote the directed subgraph of D induced by W. Throughout this paper, we assume that D is acyclic.

Dynamic network. We denote by $\mathcal{N} = (D = (V, A), c, \tau, b, s)$ a dynamic network $\mathcal{N}$ which consists of the directed graph $D = (V, A)$, a capacity function $c \colon A \to \mathbb{R}_+$ which represents the upper bound for the rate of flow that enters an arc per unit time, a transit time function $\tau \colon A \to \mathbb{Z}_+$ which represents the time required to traverse an arc, a supply function $b \colon V \to \mathbb{R}_+$ which represents the supply of a vertex, and a single sink $s \in V$. In order to avoid complicated argument, we assume $\tau(e) > 0$ for any $e \in A$. In this paper, we use the following notations: (i) $c(W_1, W_2) = \sum_{e \in \delta(W_1, W_2)} c(e)$ for any $W_1, W_2 \subseteq V$, and (ii) $c(W) = c(W, V - W)$ and $b(W) = \sum_{v \in W} b(v)$ for any $W \subseteq V$. Since we consider evacuation to s, we assume that s has no leaving arcs and no supply, and any vertex is reachable to s. We define a *length* of a path p in D as $\sum_{e \in p} \tau(e)$. We define a *dynamic network flow* $f \colon A \times \mathbb{Z}_+ \to \mathbb{R}_+$ in $\mathcal{N}$ as follows. For any $e \in A$ and $\theta \in \mathbb{Z}_+$, we denote by $f(e, \theta)$ the flow rate entering e at the time step θ which arrives at the head of e at the time step $\theta + \tau(e)$. We call f a *feasible dynamic network flow* in $\mathcal{N}$ if it satisfies

[1] In order to distinguish classical network from dynamic network, we call classical network *static network*.

the following three conditions, i.e., capacity constraint **CC**, flow conservation **FC**, and demand constraint **DC** [4].
CC: For any $e \in A$ and $\theta \in \mathbb{Z}_+$, $0 \leq f(e,\theta) \leq c(e)$.
FC: For any $v \in V$ and $\Theta \in \mathbb{Z}_+$,

$$\sum_{e \in \delta^+(v)} \sum_{\theta=0}^{\Theta} f(e,\theta) - \sum_{e \in \delta^-(v)} \sum_{\theta=0}^{\Theta-\tau(e)} f(e,\theta) \leq b(v).$$

DC: There exists $\Theta \in \mathbb{Z}_+$ such that

$$\sum_{e \in \delta^-(s)} \sum_{\theta=0}^{\Theta-\tau(e)} f(e,\theta) = \sum_{v \in V} b(v). \tag{1}$$

For a feasible dynamic network flow f, let $\Theta(f)$ denote the minimum time step Θ satisfying (1). The *evacuation problem* asks to find the minimum value of $\Theta(f)$ among all feasible dynamic network flows f. Given a dynamic network $\mathcal{N}$, the evacuation problem EP($\mathcal{N}$) is formally defined as follows:

$$\text{EP}(\mathcal{N})\colon\ \text{minimize}\,\{\Theta(f)\colon f \text{ is a feasible dynamic network flow in } \mathcal{N}\}\,.$$

We define $\Theta(\mathcal{N})$ as the optimal value of EP($\mathcal{N}$). Given time horizon T, we define the *decision version of EP($\mathcal{N}$) with time horizon* T as the problem which determines whether there exists a feasible dynamic network flow f with $\Theta(f) \leq T$ in $\mathcal{N}$. Throughout this paper, n and m denote $|V|$ and $|A|$, respectively.
Static network. We denote by $\mathcal{N}' = (D' = (V', A'), c', b', s')$ a *static network* $\mathcal{N}'$ which consists of the directed graph $D' = (V', A')$, a capacity function $c'\colon A' \to \mathbb{R}_+$ a supply function $b'\colon V' \to \mathbb{R}_+$, and a single sink $s' \in V'$. We call $f\colon A' \to \mathbb{R}_+$ a *feasible static network flow* in $\mathcal{N}'$ if it satisfies the following two conditions, i.e., capacity constraint **CC** and flow conservation **FC**.
CC: For any $e \in A'$, $0 \leq f(e) \leq c'(e)$.
FC: For any $v \in V' - s'$, $\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta^-(v)} f(e) = b'(v)$.
If there exists a feasible flow in $\mathcal{N}'$, $\mathcal{N}'$ is called *feasible*.
Time-expanded network. To solve the decision version of EP($\mathcal{N}$) with time horizon T, Ford and Fulkerson [3] introduced the *time-expanded network* which is a static network such that for any $v \in V$ and $i = 0, 1, \ldots, T$, there is a vertex v_i, and for any $e = uv \in A$, $i = 0, 1, \ldots, T - \tau(e)$, there is an arc $e_i = u_i v_{i+\tau(e)}$ whose capacity is $c(e)$, and for any $v \in V$ and $i = 0, 1, \ldots, T-1$, there is a *holdover arc* $v_i v_{i+1}$ with infinite capacity. For any $v \in V$ the supply of v_0 is set to $b(v)$ and the supplies of all the other vertices v_i for $i = 1, \ldots, T$ are set to zero. Let s_T be a sink in the time-expanded network (Fig. 1). Though we can decide whether the time-expanded network is feasible or not by solving the maximum-flow problem, the running time is pseudo-polynomial because the size of the time-expanded network is pseudo-polynomial in the input size.

2.1 Dynamic Networks with Uniform Path-Lengths

From here, we assume that any dynamic network satisfies uniform path-length condition. First we review the result due to Hall et al. [5]. They proved that EP($\mathcal{N}$) can be reduced to the *parametric flow problem* defined on the *condensed time-expanded network* whose size is polynomial in the input size.

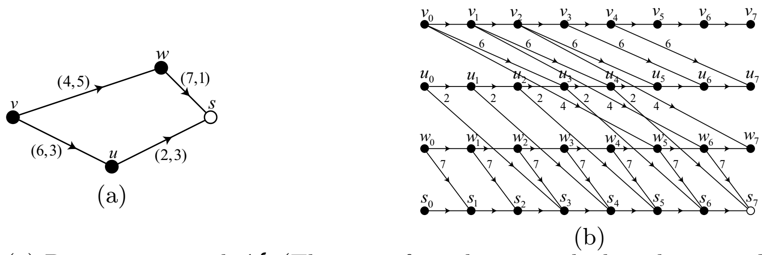


Fig. 1. (a) Dynamic network $\mathcal{N}$. (The pair of numbers attached to the arc indicates the capacity and the transit time.) (b) Time-expanded network with $T = 7$. (The number attached to the arc indicates the capacity.)

We introduce necessary notations for $\mathcal{N} = (D = (V, A), c, \tau, b, s)$. For $v \in V$, we define l_v as the length of a path from v to s. Let us arrange the distinct values in $\{l_v : v \in V\}$ as $L_1 < \cdots < L_k$ where $L_1 = 0$ and k is the number of the distinct path-lengths in $\mathcal{N}$. Without loss of generality we assume that for any i with $2 \le i \le k$ $b(v) > 0$ for at least one vertex $v \in V$ with $l_v = L_i$. Let $L_{k+1} = T + 1$. We say a vertex v is at level i when $l_v = L_i$, which is denoted by $lev(v) = i$. We partition interval $[0, T]$ into $I_1, I_2, \ldots, I_k$ such that $I_i = [L_i, L_{i+1} - 1]$ holds for $i = 1, \ldots, k$. Moreover, let $P_s = \{v \in V : e = vs \in A\}$ and $R_v = \{w \in P_s : w \text{ is reachable from } v \text{ in } D\}$ for $v \in V$. For example, for $\mathcal{N}$ in Fig. 1(a) with $T = 7$, we obtain $(l_s, l_w, l_u, l_v) = (0, 1, 3, 6)$. Thus, we have $k = 4$ and $I_1 = \{0\}$, $I_2 = \{1, 2\}$, $I_3 = \{3, 4, 5\}$, $I_4 = \{6, 7\}$.

The *condensed time-expanded network* $\mathcal{N}^c = (D^c = (V^c, A^c), c^c, b^c, s^c)$ for $\mathcal{N}$ with time horizon T is defined as follows. V^c is defined as $\{v_i : v \in V, i = lev(v), \ldots, k\}$. A^c consists of two types, i.e., $A^c = A_1^c \cup A_2^c$. $A_1^c = \{e_i = u_i v_i : e = uv \in A, i = lev(u), \ldots, k\}$ and $A_2^c = \{v_i v_{i+1} : v \in V, i = lev(v), \ldots, k - 1\}$. Arc $e_i \in A_1^c$ has the capacity $|I_i| c(e)$ where $|I_i|$ denotes the number of elements in I_i. An arc in A_2^c is a holdover arc whose capacity is infinity. For $v \in V$ the supply of $v_{lev(v)}$ is set to $b(v)$ and the supplies of all the other vertices v_i for $i = lev(v) + 1, \ldots, k$ are set to zero. $s^c = s_k$ holds (Fig. 2(a)).

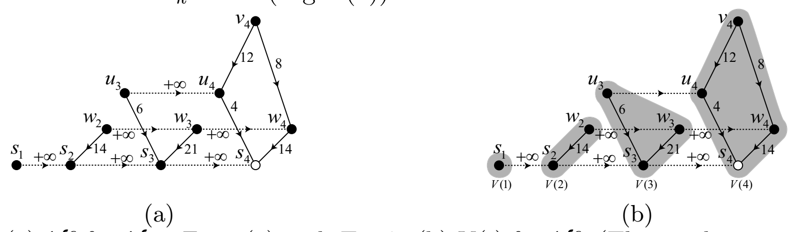


Fig. 2. (a) $\mathcal{N}^c$ for $\mathcal{N}$ in Fig. 1(a) with $T = 7$. (b) $V(i)$ for $\mathcal{N}^c$. (The number attached to the arc indicates the capacity, and holdover arcs are illustrated by dotted lines.)

For $i = 1, \ldots, k$, let $V(i) = \{v_i \in V^c : v \in V\}$ and $A(i) = \{e_i \in A_1^c : e \in A\}$. Notice that $V(i)$ for $i = 1, \ldots, k$ partitions V^c. It is easy to see that $A(i)$ is the arc set of $D^c[V(i)]$, i.e., the subgraph of D^c induced by $V(i)$ (Fig. 2(b)). From the definition of $\mathcal{N}^c$, we have the following fact.

Fact 1 For any $i, j = 1, \ldots, k$ with $j - i \neq 1$, there is no arc connecting from $V(i)$ to $V(j)$. For any $i = 1, \ldots, k-1$, $\delta(V(i), V(i+1)) = \{v_i v_{i+1} : v_i \in V(i)\}$ holds.

From Fact 1, we can see that (i) $\mathcal{N}^c$ consists of k components such that for any $i = 1, \ldots, k$, the *i-th component* is a directed graph $D^c[V(i)]$ such that capacity of $e_i \in A(i)$ is $|I_i|c(e)$ (Fig. 2(b)), and (ii) consecutive components are connected by holdover arcs. Let $V_{\leq i} = \{v \in V : lev(v) \leq i\}$ for $i = 1, \ldots, k$.

Lemma 1. *(i) For any $i = 1, \ldots, k$, $D^c[V(i)]$ is isomorphic to $D[V_{\leq i}]$. (ii) For any $i = 1, \ldots, k$ and $u, v \in V_{\leq i}$, $\lambda_{D^c[V(i)]}(u_i, v_i) = \lambda_D(u, v)$.*

Proof. (i) follows from the definition of $D^c[V(i)]$. (ii) follows from $\lambda_{D[V_{\leq i}]}(u, v) = \lambda_D(u, v)$ for $i = 1, \ldots, k$ and $u, v \in V_{\leq i}$ and from (i). □

Hall et al. showed that a feasible dynamic flow f with $\Theta(f) \leq T$ exists in $\mathcal{N}$ if and only if $\mathcal{N}^c$ is feasible for time horizon T. Thus EP($\mathcal{N}$) can be solved by computing the minimum T such that $\mathcal{N}^c$ is feasible. By regarding T as the parameter we can reduce EP($\mathcal{N}$) to the *parametric flow problem* defined as follows.
Parametric flow problem. Given a static network $\mathcal{N}' = (D' = (V', A'), c', b', s')$ such that the capacity of $e \in A'$ is represented by $a_e + \xi g_e$ where a_e is a real constant, g_e is a nonnegative constant, and ξ is a nonnegative parameter, the parametric flow problem asks to find the minimum value of ξ such that $\mathcal{N}'$ is feasible. This problem can be solved in $O(|A'|^2|V'| \log(|V'|^2/|A'|))$ time by using the algorithm of [8].

Notice that from $L_{k+1} = T + 1$ $c^c(e_k) = |I_k|c(e) = (T - L_k + 1)c(e)$. The following theorem follows from $|V^c| = O(kn)$ and $|A^c| = O(km)$.

Lemma 2 ([5]). *EP($\mathcal{N}$) can be solved in $O(k^3 m^2 n \log(kn^2/m))$ time.*

3 Evacuation Problem for a Fully Connected Network

A dynamic network $\mathcal{N} = (D = (V, A), c, \tau, b, s)$ is called *fully connected* if for each vertex $v \in V - s$ the minimum v-s cut is determined by the arcs incident to s whose tails are reachable from v. That is, the value of the minimum v-s cut is equal to $\sum_{e \in \delta(R_v, s)} c(e)$. In the subsequent discussion, we concentrate on the unit capacity case, i.e., the capacity of every arc is equal to one. In this case, $\mathcal{N}$ is fully connected if and only if $\lambda_D(v, s) = |\delta(R_v, s)|$ holds for any $v \in V - s$. The general capacity case can be treated similarly, and we will consider the general capacity case at the end of this section. In this section, we prove that EP($\mathcal{N}$) for a fully connected network can be solved efficiently. This is a generalization of the result of [2]. We will prove that the problem can be reduced to to the *restricted parametric flow problem* defined in Section 3.2.

For the subsequent discussion, we will define *contraction* in a static network $\mathcal{N}' = (D' = (V', A'), c', b', s')$ and show the sufficient condition such that we can contract some vertex set in $\mathcal{N}^c$. The contraction of $X \subseteq V' - s'$ in $\mathcal{N}'$ is defined as the operation which consists of shrinking the vertices in X into a single vertex, eliminating loops, and combining multiple arcs by adding their capacities. For $X \subseteq V' - s'$, we call X *contractible* when $\mathcal{N}'_{/X}$ is feasible if and only if $\mathcal{N}'$ is feasible. We then give the sufficient condition such that X is contractible in $\mathcal{N}'$.

Lemma 3. *For $X \subseteq V' - s'$, if there exists $Y \subseteq V' - s'$ with $X \subseteq Y$ such that $c'(Z) \geq c'(Y \cup Z)$ holds for any $Z \subseteq V' - s'$ with $X \cap Z \neq \emptyset$ and $X \not\subseteq Z$, X is contractible. (See the appendix for the proof of the lemma.)*

3.1 Contraction in the condensed time-expanded network

For $\mathcal{N} = (D = (V, A), c, \tau, b, s)$ and $Q \subseteq P_s$, let $\mathcal{P}_Q = \{v \in V : R_v \subseteq Q\}$ and $\mathcal{P}^*_Q = \{v \in V : R_v = Q, \lambda_D(v, s) = |\delta(Q, s)|\}$ (Fig. 3(a)). If $\mathcal{N}$ is fully connected, $V - s = \bigcup_{Q \subseteq P_s} \mathcal{P}^*_Q$ holds. For any $W \subseteq V$ and $i = 1, \ldots, k$, let $W(i) = \{v_i \in V^c : v \in W\}$. The following theorem will be used in the subsequent discussion.

Theorem 1. *For any $Q \subseteq P_s$ and $i = 1, 2, \ldots, k$, $\mathcal{P}^*_Q(i)$ is contractible in $\mathcal{N}^c$.*

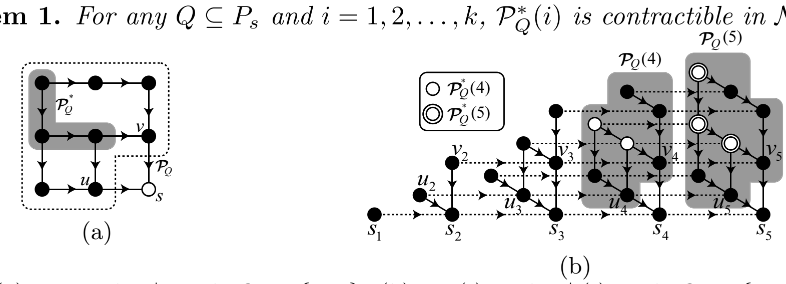


Fig. 3. (a) $\mathcal{P}_Q$ and $\mathcal{P}^*_Q$ with $Q = \{u, v\}$. (b) $\mathcal{P}_Q(i)$ and $\mathcal{P}^*_Q(i)$ with $Q = \{u, v\}$ and $i = 4, 5$. (The transit time in Fig 3(a) takes the same value.)

The following lemma will be used in the proof of Theorem 1. The proof is given in appendix.

Lemma 4. *$\delta^+(\bigcup_{i \leq j \leq k} \mathcal{P}_Q(j)) = \bigcup_{i \leq j \leq k} \delta(Q(j), s_j)$ holds.*

Proof. (**Theorem 1**) Let us fix $Q \subseteq P_s$ and i. We will use Lemma 3 to prove that $\mathcal{P}^*_Q(i)$ is contractible by setting $X = \mathcal{P}^*_Q(i)$ and $Y = \bigcup_{i \leq j \leq k} \mathcal{P}_Q(j)$. Thus, it is sufficient to prove $c^c(Z) \geq c^c(Y \cup Z)$ for any $Z \subseteq V^c - s^c$ with $X \cap Z \neq \emptyset$ and $X \not\subseteq Z$. In order to prove $c^c(Z) \geq c^c(Y \cup Z)$, it is sufficient to prove $c^c(Y) \leq c^c(Y \cap Z)$ since c^c is a submodular function. Recalling that every arc capacity is assumed to be one, $c^c(e_j) = |I_j|$ holds. Thus from Lemma 4, we have

$$c^c(Y) = \textstyle\sum_{j=i}^{k} |\delta(Q(j), s_j)||I_j|. \tag{2}$$

Now we evaluate $c^c(Y \cap Z)$. From $X \cap Z \neq \emptyset$, let $v^*_i \in X \cap Z$. Since the capacity of holdover arc is infinity, we can assume $v^*_j \in Z$ holds for any $j = i+1, \ldots, k$ since otherwise $c^c(Y \cap Z) = +\infty$ and the theorem clearly holds. We have

$$c^c(Y \cap Z) \geq \textstyle\sum_{j=i}^{k} \sum_{e \in \delta(\mathcal{P}_Q(j) \cap Z, V(j) - (\mathcal{P}_Q(j) \cap Z))} c^c(e) \tag{3}$$

(See the appendix for the proof of this inequality). Since $\delta(\mathcal{P}_Q(j) \cap Z, V(j) - (\mathcal{P}_Q(j) \cap Z))$ is the set of arcs outgoing from $\mathcal{P}_Q(j) \cap Z$ in the j-th component, the following inequality holds for every j with $j = i, i+1, \ldots, k$

$$\begin{aligned} \textstyle\sum_{e \in \delta(\mathcal{P}_Q(j) \cap Z, V(j) - (\mathcal{P}_Q(j) \cap Z))} c^c(e) &\geq \lambda_{D^c[V(j)]}(v^*_j, s_j)|I_j| \quad (\text{from } v^*_j \in \mathcal{P}_Q(j) \cap Z) \\ &= \lambda_D(v^*, s)|I_j| \quad (\text{from Lemma 1(ii)}) = |\delta(Q, s)||I_j| \; (\text{from } v^* \in \mathcal{P}^*_Q). \end{aligned} \tag{4}$$

Since $|\delta(Q(j), s_j)| \leq |\delta(Q, s)|$ holds, we have from (2), (3) and (4)

$$c^c(Y) = \textstyle\sum_{j=i}^{k} |\delta(Q(j), s_j)||I_j| \leq \sum_{j=i}^{k} |\delta(Q, s)||I_j| \leq c^c(Y \cap Z). \qquad \square$$

3.2 The restricted parametric flow problem

In this problem, we are given a static network with multiple sinks $\mathcal{N}'' = (D'' = (V'', A''), c'', b'', S'')$ such that (i) S'' is a set of sinks, (ii) the capacity $c''(e)$ for an arc e incident to a sink is a linear function $a_e + g_e\xi$ where a_e is a constant, g_e is a nonnegative constant and ξ is a nonnegative parameter. The problem asks to find the minimum value of ξ such that $\mathcal{N}''$ is feasible where we define $f\colon A'' \to \mathbb{R}_+$ a feasible flow in $\mathcal{N}''$ when it satisfies **CC** and **FC** for any $v \in V'' - S''$. This problem can be transformed into a parametric maximum-flow problem studied by [7] by introducing a super source vertex q and arcs from q to every vertex v with $b''(v) > 0$ such that the capacity of qv is set to $b''(v)$. It is then easy to see that $\mathcal{N}''$ is feasible for a fixed ξ if and only if the maximum-flow value from q to S'' in the transformed problem is at least $\sum_{v\in V''} b''(v)$. Regarding ξ as a parameter, the maximum-flow value is a linear function in ξ.

Lemma 5 ([7]). *The maximum-flow value from q to S'' in the transformed network is a non-decreasing piecewise linear concave function $\kappa(\xi)$, and the largest breakpoint of $\kappa(\xi)$ can be found in the same time complexity as that of a single computation of the maximum-flow, i.e., $O(|A''||V''|\log(|V''|^2/|A''|))$.*

Lemma 6. *We can determine whether there exists ξ such that $\mathcal{N}''$ is feasible, and if there exists such ξ, the minimum such value can be found in $O(|A''||V''|\log(|V''|^2/|A''|))$.*

Proof. From the above discussion, $\mathcal{N}''$ is feasible when there exists ξ such that maximum-flow value in the transformed problem is larger than or equal to $\sum_{v\in V''} b''(v)$. On the other hand, the maximum-flow value in the transformed problem can not exceed $\sum_{v\in V''} b''(v)$. The slope of $\kappa(\xi)$ is zero and $\kappa(\xi)$ is less than or equal to $\sum_{v\in V''} b''(v)$ when ξ is larger than the largest breakpoint. Checking whether there exists ξ such that $\mathcal{N}''$ is feasible reduces to computing the largest breakpoint of $\kappa(\xi)$. Moreover, if there exists ξ such that $\mathcal{N}''$ is feasible, the minimum value of ξ such that $\mathcal{N}''$ is feasible is equal to the largest breakpoint of $\kappa(\xi)$. Thus, the lemma follows from Lemma 5. □

As was defined in Section 2.1, in the condensed time-expanded network, the capacity of all arcs in the k-th component $D^c[V(k)]$ contains the parameter T, i.e., linear function of T. In Fig. 2(a), regarding T as the parameter, we have $c^c(u_4s_4) = 2(T-5)$, $c^c(w_4s_4) = 7(T-5)$, $c^c(v_4u_4) = 6(T-5)$, and $c^c(v_4w_4) = 4(T-5)$. Thus, the arcs which are not incident to a sink (i.e., v_4u_4 and v_4w_4) have the parametric capacity. Therefore, we can not reduce $\mathrm{EP}(\mathcal{N})$ for a general dynamic network with uniform path-lengths to the restricted parametric flow problem.

3.3 Reduction to the restricted parametric flow problem

Our reduction is constructed by the following lemmas. First, given a vertex set $\hat{V}$, a supply function $\hat{b}\colon \hat{V} \to \mathbb{R}_+$, a path-length function $\hat{l}\colon \hat{V} \to \mathbb{R}_+$, and a sink $\hat{s} \in \hat{V}$, let $\boldsymbol{\mathcal{N}}(\hat{V}, \hat{b}, \hat{l}, \hat{s})$ be a set of dynamic networks $\hat{\mathcal{N}} = (\hat{D} = (\hat{V}, \hat{A}), \hat{c}, \hat{\tau}, \hat{b}, \hat{s})$ which satisfies (i) $|\delta^-(\hat{s})| = 1$, (ii) for any $v \in \hat{V}$ the length from v to $\hat{s}$ is equal to $\hat{l}_v$, and (iii) $\hat{c}(e) = 1$ for any $e \in \hat{A}$. Since we are only given path-length function but not the arc set or transit time of arcs, there may exist many possible networks which satisfy the given path-length function. For example, given $\hat{V} = \{\hat{s}, x, y, v, w\}$, $(\hat{b}(\hat{s}), \hat{b}(x), \hat{b}(y), \hat{b}(v), \hat{b}(w)) = (0, 4, 3, 5, 1)$, and $(\hat{l}_{\hat{s}}, \hat{l}_x, \hat{l}_y, \hat{l}_v, \hat{l}_w) = (0, 1, 2, 4, 7)$, all dynamic networks in Fig. 4 belong to $\boldsymbol{\mathcal{N}}(\hat{V}, \hat{b}, \hat{l}, \hat{s})$.

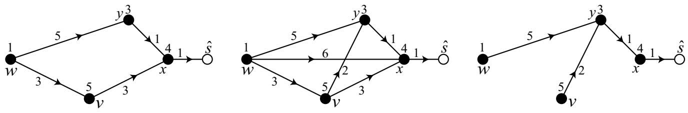


Fig. 4. Example of dynamic networks in $\boldsymbol{\mathcal{N}}(\hat{V}, \hat{b}, \hat{l}, \hat{s})$. (The numbers attached to the arc and the vertex indicate the capacity and the supply, respectively.)

Lemma 7. *For any $\hat{\mathcal{N}} \in \boldsymbol{\mathcal{N}}(\hat{V}, \hat{b}, \hat{l}, \hat{s})$, $\Theta(\hat{\mathcal{N}})$ takes the same value regardless of the underlying network topology of $\hat{\mathcal{N}}$.*

Proof. For any $\hat{\mathcal{N}} = (\hat{D} = (\hat{V}, \hat{A}), \hat{c}, \hat{\tau}, \hat{b}, \hat{s}) \in \boldsymbol{\mathcal{N}}(\hat{V}, \hat{b}, \hat{l}, \hat{s})$, $P_{\hat{s}}$ consists of a single element from $|\delta^-(\hat{s})| = 1$. Thus, $\hat{\mathcal{N}}$ is fully connected because any $v \in \hat{V}$ is reachable to $\hat{s}$ by using the path of length $\hat{l}_v$. Since $\hat{l}_v$ does not depend on the choice of $\hat{\mathcal{N}}$, the number of distinct values in $\{\hat{l}_v : v \in \hat{V}\}$ does not depend on the choice of $\hat{\mathcal{N}}$. Let $\hat{k}$ denote this number. Let $\hat{\mathcal{N}}^c$ be the condensed time-expanded network for $\hat{\mathcal{N}}$. Since $\hat{\mathcal{N}}$ is fully connected, $\hat{V}(i) - \hat{s}_i$ is contractible in $\hat{\mathcal{N}}^c$ for any $i = 1, \ldots, \hat{k}$ from Theorem 1. Let $\hat{\mathcal{N}}^* = (\hat{D}^* = (\hat{V}^*, \hat{A}^*), \hat{c}^*, \hat{b}^*, \hat{s}^*)$ be the one obtained by contracting $\hat{V}(i) - \hat{s}_i$ into a single vertex p_i for every $i = 1, \ldots, \hat{k}$ in $\hat{\mathcal{N}}^c$. It is easy to see that arcs whose capacity is not infinity in $\hat{\mathcal{N}}^*$ are $p_i \hat{s}_i$ with $i = 1, \ldots, \hat{k}$ and the capacity of $p_i \hat{s}_i$ is equal to $|\hat{I}_i|$ since the capacity of any arc is assumed to be one where $\hat{I}_i$ is defined for $\hat{\mathcal{N}}$ in a manner similar to I_i for $\mathcal{N}$. It is easy to see that $|\hat{I}_i|$ does not depend on the choice of $\hat{\mathcal{N}}$ from the definition of $\hat{I}_i$. Since $\hat{V}(i)$ does not depend the choice of $\hat{\mathcal{N}}$, the supply of p_i does not depend on the choice of $\hat{\mathcal{N}}$. From the above discussion, regardless of the choice of $\hat{\mathcal{N}} \in \boldsymbol{\mathcal{N}}(\hat{V}, \hat{b}, \hat{l}, \hat{s})$, $\hat{\mathcal{N}}^*$ is the same. This completes the proof. □

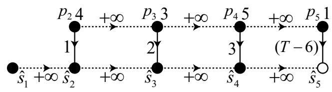


Fig. 5. $\hat{\mathcal{N}}^*$ for dynamic network in Fig. 4. (The numbers attached to the vertex and the arc indicate the supply and the capacity, respectively.)

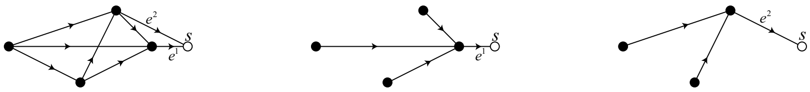
Form the proof of Lemma 7, we can see that for any $\hat{\mathcal{N}} \in \boldsymbol{\mathcal{N}}(\hat{V}, \hat{b}, \hat{l}, \hat{s})$ $\Theta(\hat{\mathcal{N}})$ depends only on the sum of the supplies of vertices $v \in \hat{V}$ such that $lev(v)$ takes the same value, but not the supply of each vertex.

For $\mathcal{N} = (D = (V, A), c, \tau, b, s)$, let $\delta(P_s, s) = \{e^1, e^2, \ldots, e^d\}$, and $V^j = \{v \in V : v \text{ is reachable to the tail of } e^j\} \cup \{s\}$.

Lemma 8. *Given a dynamic network $\mathcal{N} = (D = (V, A), c, \tau, b, s)$, there exist d arc-disjoint s-rooted trees $D^j = (V^j, A^j)$ for $j = 1, \ldots, d$ such that D^j spans V^j and $A^j \subseteq A$ if and only if $\lambda_D(v, s) = |\delta(R_v, s)|$ holds for any $v \in V - s$.*

Fig. 6(b) and (c) illustrate D^1 and D^2 of the directed graph D in Fig. 6(a).

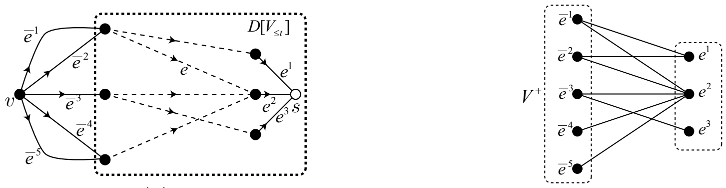
Proof. It is not difficult to see that "only if-part" holds. We then prove the "if-part". We prove that there exist d s-rooted trees satisfying the lemma statement



(a) (b) (c)

Fig. 6. (a) $D = (V, A)$, (b) $D^1 = (V^1, A^1)$, (c) $D^2 = (V^2, A^2)$.

by induction on $i = 2, \dots, k$ for $D[V_{\leq i}]$. For $i = 2$, this lemma clearly holds. Assuming that the lemma holds for the induced subgraph $D[V_{\leq t}]$ with $t \geq 2$, we will prove the lemma also holds for $D[V_{\leq t+1}]$. For an arbitrary $v \in V_{\leq t+1} - V_{\leq t}$ we define the bipartite graph $G = ((V^+, V^-), E)$ as follows. Let V^+ and V^- represent the set of arcs whose tail is v and those whose tail belongs to R_v, respectively. $v^+ \in V^+$ and $v^- \in V^-$ are joined by an edge in E if and only if the head of the arc which corresponds to v^+ is reachable to the tail of the arc which corresponds to v^- (Fig. 7).



(a) (b)

Fig. 7. (a) Arcs between $V_{\leq t}$ and $v \in V_{\leq t+1} - V_{\leq t}$. (The arc e represented by a dotted line means that the tail of e is reachable to the head of e.) (b) G for v in (a).

In order to prove that the lemma holds for $D[V_{\leq t} \cup v]$, we will show that there always exists a matching $\mathcal{M}$ which saturates V^- in G. This is because if there exists such $\mathcal{M}$, from the induction hypothesis we can extend arc-disjoint s-rooted trees satisfying the lemma in $D[V_{\leq t}]$ to the ones in $D[V_{\leq t} \cup v]$. Let $e = v^+ v^- \in \mathcal{M}$, and let $\bar{e}^i$ and e^j be arcs corresponding to v^+ and v^-, respectively. Let $\mathcal{T}^j$ denote the s-rooted tree containing e^j which satisfies the induction hypothesis. $\mathcal{T}^j$ can be extended by including $\bar{e}^i$. Then s-rooted trees so extended for all $e \in \mathcal{M}$ become also arc-disjoint. In order to prove the existence of $\mathcal{M}$, we use Hall's theorem [9]. Assume that there exists some $W \subseteq V^-$ with $|W| > |\mathrm{Ne}(W)|$ where $\mathrm{Ne}(W)$ is the set of vertices adjacent to some element of W. This contradicts the fact that there exist $|\delta(R_v, s)|$ arc-disjoint paths from v to s. This is because that the paths contain the arcs corresponding to W have to contain the arcs corresponding to $\mathrm{Ne}(W)$, and thus these paths are not arc-disjoint from $|W| > |\mathrm{Ne}(W)|$. $\square$

Now let us fix $\{b^j : j = 1, \dots, d\}$ such that (i) For any $v \in V$, $\sum_{j=1}^{d} b^j(v) = b(v)$ holds, and (ii) for any $v \in V$ and $j = 1, \dots, d$ with $v \notin V^j$, $b^j(v) = 0$ holds. Intuitively speaking, $b^j(v)$ represents the assignment of the supply of v which reaches s through $D^j = (V^j, A^j)$. For a fully connected network $\mathcal{N} = (D = (V, A), c, \tau, b, s)$, let $\mathcal{N}^j = (D^j = (V^j, A^j), c^j, \tau^j, b^j, s)$ where c^j and τ^j respectively denote c and τ whose domain is restricted to A^j. Notice that from Lemma 7 $\Theta(\mathcal{N}^j)$ does not depend on the choice of A^j if b^j is fixed. Let f^j_{opt} be an optimal dynamic network flow in $\mathcal{N}^j$. Recalling that since $A^{j_1} \cap A^{j_2} = \emptyset$ holds with $j_1 \neq j_2$, the dynamic flow obtained by combining f^j_{opt} for all $j = 1, \ldots, d$ is feasible in $\mathcal{N}$.

Lemma 9. *Given a fully connected network $\mathcal{N} = (D = (V, A), c, \tau, b, s)$, under the constraint such that for each $v \in V$ the amount of $b(v)$ which reaches s through e^j is $b^j(v)$, $\Theta(\mathcal{N})$ is equal to* $\max\{\Theta(\mathcal{N}^j) \colon j = 1, \ldots, d\}$.

The proof of this lemma is almost the same as Theorem 3 in [2], and hence is given in appendix. From Lemma 9, we only need to determine b^j for $j = 1, \ldots, d$ to obtain $\Theta(\mathcal{N})$.

Lemma 10. *We can reduce EP($\mathcal{N}$) for a fully connected network $\mathcal{N}$ to the restricted parametric flow problem.*

We will prove the lemma as follows.

For a fully connected network $\mathcal{N} = (D = (V, A), c, \tau, b, s)$, let $\mathcal{R}(\mathcal{N}) = (D_R = (V_R, A_R), c_R, b_R, S_R)$ be the static network with multiple sinks to which EP($\mathcal{N}$) is reduced. First we consider $\mathcal{R}(\mathcal{N})$ in the case of $|\delta^-(s)| = 1$. In this case, $\mathcal{R}(\mathcal{N})$ is the same as $\hat{\mathcal{N}}^*$ defined in the proof of Lemma 7. Notice that the parameter T is contained only in the capacity of the arc which is incident to a sink $\hat{s}_{\hat{k}}$ by the definition of $\hat{I}_i$ (e.g. see Fig. 5). It is clear that in order to compute $\Theta(\mathcal{N})$ we need to compute T^* which is the minimum value of T such that $\mathcal{R}(\mathcal{N})$ is feasible, i.e., the solution of the restricted parametric flow problem defined on $\mathcal{R}(\mathcal{N})$. Notice that $\Theta(\mathcal{N}) = \lceil T^* \rceil$ holds.

From the above discussion, we can construct $\mathcal{R}(\mathcal{N})$ for the case of $|\delta^-(s)| > 1$ in three steps as follows. $\mathcal{R}(\mathcal{N})$ is constructed so that the minimum value of $\max\{\Theta(\mathcal{N}^j) \colon j = 1, \ldots, d\}$ among all b^j with $j = 1, \ldots, d$ is equal to $\lceil T^* \rceil$ where T^* is the same as defined above and we can compute an optimal allocation of the supplies b^j with $j = 1, \ldots, d$ which attains T^*, i.e. $\Theta(\mathcal{N})$. Let $V(i, Q) = \{v \in V \colon lev(v) = i, R_v = Q\}$.

(i) We first construct *gadget* G^j separately for each $j = 1, \ldots, d$ which is the same as $\mathcal{R}(\mathcal{N}^j)$ with no supply (Fig. 8(a), (b), and (c)). Notice that the parameter T is common in all gadgets. (ii) For every nonempty $V(i, Q)$, we add vertices u_i^Q in V_R. The supply of u_i^Q (denoted by $b_R(u_i^Q)$) is defined as the sum of supplies in $V(i, Q)$. (iii) We add the arc from u_i^Q to the gadget G^j in A_R if $V^j \cap V(i, Q) \neq \emptyset$. Notice that the allocation of the supply of u_i^Q to the gadget G^j means that we allocate the supplies of $V(i, Q)$ to $\mathcal{N}^j$. We determine to which vertex in G^j u_i^Q is connected as follows. For any $j = 1, \ldots, d$, we arrange the distinct values $\{l_v \colon v \in V^j\}$ as $L^j_1 < \cdots < L^j_{k^j}$. We connect u_i^Q to $p_{i'}$ in G^j with $L^j_{i'} = L_i$. Notice that from the way of construction of $\mathcal{R}(\mathcal{N})$ the parameter T is contained only in the capacity of the arc which is incident to s_{k^j} in each gadget G^j. Therefore, all arcs in A_R whose capacity contains the parameter T are incident to sinks S_R. Lemma 10 then follows from the way of construction of $\mathcal{R}(\mathcal{N})$. For example, in step(ii) $u_4^{\{x,y\}}$ in Fig. 8(d) is added to allocate the supply of v in Fig. 8(a). In step(iii), for $\mathcal{N}^1$ and $\mathcal{N}^2$ in Fig. 8(b), $k^1 = 4$ and $k^2 = 3$ hold, and $u_4^{\{x,y\}}$ in Fig. 8(d) is connected to p_4 in G^1 and p_3 in G^2. In Fig. 8(c) and (d), only $p_4 s_4$ in G^1 and $p_3 s_3$ in G^2 contain the parameter T.

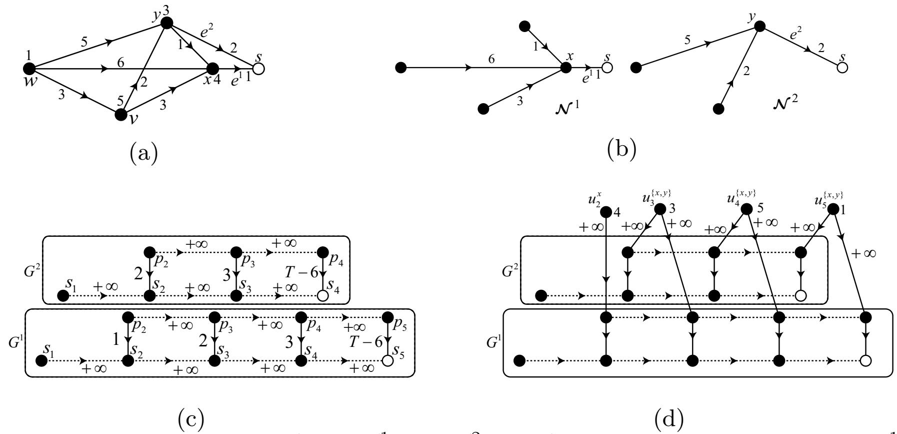


Fig. 8. (a) Dynamic network $\mathcal{N}$. (b) $\mathcal{N}^1$ and $\mathcal{N}^2$ for $\mathcal{N}$ with no supply. (c) Gadgets G^1 and G^2. (d) Vertices and arcs introduced to allocate supplies.

As was seen in Section 3.2, the restricted parametric flow problem defined on $\mathcal{R}(\mathcal{N})$ can be transformed into the parametric maximum-flow problem studied by [7] by adding the super source vertex q as well as arcs from q to all u_i^Q's in V_R such that the capacity of qu_i^Q is set to $b_R(u_i^Q)$. Since in this parametric maximum-flow problem the capacities of all cuts except $\delta(q, V_R)$ diverge to ∞ from the way of construction of $\mathcal{R}(\mathcal{N})$ as T goes to ∞, the maximum flow value of the parametric maximum-flow problem is bounded by $\sum_{v \in V_R} b_R(v)$, i.e., there always exists T such that $\mathcal{R}(\mathcal{N})$ is feasible. Since we assume $b(v) > 0$ for at least one vertex v with $lev(v) = k$, $\Theta(\mathcal{N}) \geq L_k$ holds. Therefore, we need to consider only the case of $T \geq L_k$ in the parametric maximum-flow problem.

3.4 Time complexity

Let η be the number of distinct combinations of the path-length from v to s and R_v, i.e., $\eta = |\{(l_v, R_v) \colon v \in V\}|$. Notice that η is equal to the number of u_i^Q defined above and $\eta = O(n)$ holds.

Theorem 2. *The evacuation problem EP($\mathcal{N}$) for a fully connected network $\mathcal{N}$ can be solved in $O(|P_s|m + n \log n + d(dk + \eta)(k + \eta) \log n)$ time.*

Proof. The term $O(|P_s|m + n \log n)$ is the time required to construct $\mathcal{R}(\mathcal{N})$ (the proof is given in appendix). The third term represents the time required to solve the restricted parametric flow problem. Let us evaluate the size of $\mathcal{R}(\mathcal{N})$. A single gadget has $O(k)$ vertices and $O(k)$ arcs. Since there exist d gadgets, the union of all gadgets has $O(dk)$ vertices and $O(dk)$ arcs. The number of vertices which is added to allocate the supplies is equal to η. The number of the arcs added to these vertices is clearly $O(d\eta)$. From the above discussion, we have $|V_R| = O(dk + \eta)$ and $|A_R| = O(dk + d\eta)$. From Lemma 6, this completes the proof. $\square$

Let us analyze the running time given in the above theorem in terms of m and n. Notice that the number of the arcs added to allocate the supplies is bounded by $O(m)$. This is because this number is at most $\sum_{v \in V-s} |\delta(R_v, s)|$ since $u_i^{R_v}$ is connected to at most $|\delta(R_v, s)|$ gadgets for $v \in V - s$ with $lev(v) = i$. Moreover, we have $\sum_{v \in V-s} |\delta(R_v, s)| \leq \sum_{v \in V-s} |\delta^+(v)| = m$ since the out-degree of v is no less than $|\delta(R_v, s)|$ from the fact that $\mathcal{N}$ is fully connected and the capacity of any arc is one. Next we prove that the union of all gadgets has $O(m)$ vertices and $O(m)$ arcs. Since $\mathcal{N}^j$ has $|V^j|$ vertices, the gadget G^j has $O(|V^j|)$ vertices and $O(|V^j|)$ arcs from the way of construction of G^j. Thus the number of vertices and arcs in the union of all gadgets are $O(\sum_{j=1}^{d} |V^j|)$, respectively. Since V^j is the union of a sink s and the set of vertices which are reachable to the tail of e^j, $\sum_{j=1}^{d} |V^j| = \sum_{v \in V-s} |R_v| + d$ holds (the term d represents the number of the copies of a sink). From $\sum_{v \in V-s} |R_v| \leq \sum_{v \in V-s} |\delta^+(v)| = m$ and $O(d) = m$, the number of vertices and arcs in the union of all gadgets are $O(m)$, respectively. Thus we have $|V_R| = O(m)$ and $|A_R| = (m)$ from $\eta = O(n)$, and the following corollary follows from Lemma 6.

Corollary 1. *The evacuation problem EP($\mathcal{N}$) for a fully connected network $\mathcal{N}$ can be solved in $O(m^2 \log n)$ time.*

If we simply apply the algorithm of [5], the time complexity is $O(k^3 m^2 n \log(kn^2/m))$. Our algorithm much improves the result of [5] in this case. In many practical cases, the in-degree of a sink can be considered as a constant. In this case, if we can regard d as a constant, the time complexity of our algorithm is $O(dm + d^2 n^2 \log n)$.
Integral capacity case. For this case, we can apply our algorithm by splitting arcs into ones whose capacity is one. In this case, we have $\mathcal{R}(\mathcal{N})$ which has $O(kn)$ vertices and $O(n^2)$ arcs by combining all gadgets corresponding to parallel arcs, and hence our algorithm can solve EP($\mathcal{N}$) in $O(kn^3 \log n)$ time. In the general capacity case, we can extend our algorithm similarly.

4 Conclusion and Remarks

In this paper, we generalize the class of networks to which the algorithm of [2] can be applied. Though the details are omitted, our algorithm can solve EP($\mathcal{N}$) for a d-dimensional grid network with uniform capacity in $O(d^2 n + n \log n + d^3 3^{2d} n^{2/d} \log n)$ time. In particular, in the case of $d = 2$, EP($\mathcal{N}$) can be solved in $O(n \log n)$ time. This time complexity matches the result of [7]. In the case where there exists a vertex v with $\lambda_D(v, s) < |\delta(R_v, s)|$ (called *deficient vertex*) in a 2-dimensional grid network with uniform capacity, this problem can be solved in $O(\sigma^3 n^{3/2} \log n)$ time by contracting the condensed time-expanded network according to Theorem 1 where σ is the number of deficient vertices.

References

1. Hoppe, B., Tardos, É.: The quickest transshipment problem. Mathematics of Operations Research **25**(1) (2000) 36–62
2. Kamiyama, N., Katoh, N., Takizawa, A.: An efficient algorithm for evacuation problems in dynamic network flows with uniform arc capacity. In: Proc. AAIM2006. Volume 4041 of LNCS., Springer (2006) 231–242
3. Ford, L., Fulkerson, D.: Flows in Networks. Princeton University Press, Princeton, NJ (1962)

4. Mamada, S., Uno, T., Makino, K., Fujishige, S.: An $O(n \log^2 n)$ algorithm for the optimal sink location problem in dynamic tree networks. Discrete Applied Mathematics **154**(16) (2006) 2387–2401
5. Hall, A., Hippler, S., Skutella, M.: Multicommodity flows over time: Efficient algorithms and complexity. In: Proc. ICALP2003. Volume 2719 of LNCS., Springer (2003) 397–409
6. Ibaraki, T., Katoh, N.: Resource allocation problems under submodular constraints. In: Resource Allocation Problems : Algorithmic Approaches. MIT Press, Cambridge, MA (1988) 144–176
7. Gallo, G., Grigoriadis, M.D., Tarjan, R.E.: A fast parametric maximum flow algorithm and applications. SIAM J. Comput. **18**(1) (1989) 30–55
8. Radzik, T.: Parametric flows, weighted means of cuts, and fractional combinatorial optimization. In Pardalos, P., ed.: Complexity in Numerical Optimization. World Scientific, River Edge, NJ (1993) 351–386
9. Pulleyblank, W.: Matchings and extensions. In Graham, R., Grötschel, M., Lovász, L., eds.: Handbook of Combinatorics. Volume 1. MIT Press (1995) 179–233
10. Hoffman, A.J.: A generalizetion of max flow - min cut. Math. Program. **6** (1974) 352–359

A Proofs

A.1 Proof of Lemma 3

Proof. It is clear that $\mathcal{N}'_{/X}$ is feasible if $\mathcal{N}'$ is feasible. Suppose that $\mathcal{N}'$ is not feasible, but $\mathcal{N}'_{/X}$ is feasible. We use the following fact.

Fact 2 ([10]) A static network $\mathcal{N}' = (D' = (V', A'), c', b', s')$ is feasible if and only if there exist no $W \subseteq V' - s'$ which satisfies $b'(W) - c'(W) > 0$.

Since we assume that $\mathcal{N}'$ is not feasible, from Fact 2 there exists some $W \subseteq V' - s'$ with $b'(W) - c'(W) > 0$.
Case 1($X \cap W = \emptyset$ hold.) It is easy to see $b'_{/X}(W) - c'_{/X}(W) = b'(W) - c'(W)$. Thus, $b'(W) - c'(W) \leq 0$ holds since $\mathcal{N}'_{/X}$ is feasible. This contradicts $b'(W) - c'(W) > 0$.
Case 2($X \subseteq W$ holds.) It is easy to see $b'_{/X}((W - X) \cup x) - c'_{/X}((W - X) \cup x) = b'(W) - c'(W)$. $b'(W) - c'(W) \leq 0$ holds since $\mathcal{N}'_{/X}$ is feasible. This contradicts $b'(W) - c'(W) > 0$.
Case 3($X \cap W \neq \emptyset$ and $X \nsubseteq W$ hold.) From $X \subseteq Y \cup W$, it is not difficult to see $b'_{/X}(((Y \cup W) - X) \cup x) - c'_{/X}(((Y \cup W) - X) \cup x) = b'(Y \cup W) - c'(Y \cup W)$. What remains is to show $b'(Y \cup W) - c'(Y \cup W) > 0$. From the statement of this lemma, there exists some $Y \subseteq V' - s'$ with $X \subseteq Y$ such that $c'(Y \cup W) \leq c'(W)$ holds. Moreover, from $W \subseteq Y \cup W$, we have $b'(Y \cup W) \geq b'(W)$, and hence $b'(Y \cup W) - c'(Y \cup W) \geq b'(W) - c'(W)$ holds. Since we have $b'(W) - c'(W) > 0$, $b'(Y \cup W) - c'(Y \cup W) > 0$ holds. This completes the proof. □

A.2 Proof of Lemma 4

Proof. The proof immediately follows from

$$\delta^{+}(\mathcal{P}_Q(j)) = \begin{cases} \delta(Q(j), s_j) \cup \{v_j v_{j+1} \colon v_j \in \mathcal{P}_Q(j)\}, & \text{if } j = i, \ldots, k-1, \\ \delta(Q(k), s_k), & \text{if } j = k. \end{cases} \quad (5)$$

(5) follows from the definition $\mathcal{P}_Q(j)$ (see Fig. 3(b)). □

A.3 Proof of (3) in Theorem 1

Proof. We have

$$\begin{aligned}\delta^{+}(\textstyle\bigcup_{i \le j \le k}(\mathcal{P}_Q(j) \cap Z)) =& \textstyle\bigcup_{i \le j \le k} \delta(\mathcal{P}_Q(j) \cap Z, V^c - \bigcup_{i \le j' \le k}(\mathcal{P}_Q(j') \cap Z)) \\ \supseteq& \textstyle\bigcup_{i \le j \le k} \delta(\mathcal{P}_Q(j) \cap Z, V(j) - (\mathcal{P}_Q(j) \cap Z)). \quad (6)\end{aligned}$$

Notice that the first equality of (6) holds since $\mathcal{P}_Q(j) \cap \mathcal{P}_Q(j') = \emptyset$ with $j \neq j'$, and the second inequality holds since $V^c - \bigcup_{i \le j' \le k}(\mathcal{P}_Q(j') \cap Z) \supseteq V(j) - (\mathcal{P}_Q(j) \cap Z)$ holds for any $j = 1, \ldots, k$. Hence we have

$$\begin{aligned} c^c(Y \cap Z) =& \textstyle\sum_{e \in \delta^{+}((\bigcup_{i \le j \le k} \mathcal{P}_Q(j)) \cap Z)} c^c(e) \\ =& \textstyle\sum_{e \in \delta^{+}(\bigcup_{i \le j \le k}(\mathcal{P}_Q(j) \cap Z))} c^c(e) \\ \ge& \textstyle\sum_{e \in \bigcup_{i \le j \le k} \delta(\mathcal{P}_Q(j) \cap Z, V(j) - (\mathcal{P}_Q(j) \cap Z))} c^c(e) \quad \text{(from (6))} \\ =& \textstyle\sum_{j=i}^{k} \sum_{e \in \delta(\mathcal{P}_Q(j) \cap Z, V(j) - (\mathcal{P}_Q(j) \cap Z))} c^c(e). \end{aligned}$$

The last equality holds since $V(j) \cap V(j') = \emptyset$ holds for $j \neq j'$. □

A.4 Proof of Lemma 9

Proof. Assume that there exists a feasible dynamic network flow f in $\mathcal{N}$ with $\Theta(f) < \max\{\Theta(\mathcal{N}^j) \colon j = 1, \ldots, d\}$. Let us decompose f into f^j with $i = 1, \ldots, d$ such that f^j represents a dynamic network flow which enter into s through e^j. For any $j = 1, \ldots, d$, let $\bar{D}^j = (V^j, \bar{A}^j)$ denote a subgraph such that $\bar{A}^j \subseteq A$ is the set of arcs e which f^j uses, i.e., $f^j(e, \theta) \ge 0$ for some $\theta \in \mathbb{Z}_{+}$. Notice that from the definition of f^j $\bar{A}^j$ contains only e^j in arcs belonging to $\delta(P_s, s)$. Let $\bar{\mathcal{N}}^j = (\bar{D}^j = (V^j, \bar{A}^j), \bar{c}^j, b^j, \bar{\tau}^j, s)$ where $\bar{c}^j$ and $\bar{\tau}^j$ are respectively c and τ whose domain is restricted to $\bar{A}^j$. Notice that f^j is a feasible dynamic network flow in $\bar{\mathcal{N}}^j$. Since $\mathcal{N}^j$ (the definition is given above Lemma 9) and $\bar{\mathcal{N}}^j$ are belong to $\boldsymbol{\mathcal{N}}(V^j, b^j, l, s)$, from Lemma 7 $\Theta(f^j) \ge \Theta(\mathcal{N}^j)$ holds. Notice that l is the path-length function of the input dynamic network $\mathcal{N}$. Thus, we have

$$\Theta(f) = \max\{\Theta(f^j) \colon j = 1, \ldots, d\} \ge \max\{\Theta(\mathcal{N}^j) \colon i = 1, \ldots, d\}.$$

It contradicts the assumption that $\Theta(f) < \max\{\Theta(\mathcal{N}^j) \colon j = 1, \ldots, d\}$. □

A.5 The rest of proof of Theorem 2

Proof. In order to construct $\mathcal{R}(\mathcal{N})$, we compute l_v for all $v \in V$ in $O(m)$ time by breadth first search from s since $\mathcal{N}$ satisfies the uniform path-length condition. After this, we can compute $lev(v)$ for all $v \in V$ in $O(n \log n)$ time by sorting $\{l_v : v \in V\}$. In order to construct $\mathcal{R}(\mathcal{N})$, we have to construct the gadgets G^j with $j = 1, \ldots, d$ and add the vertices and arcs to allocate the supplies. In order to construct the gadgets, we have to obtain k^j and $\{L_1^j, \ldots, L_{k^j}^j\}$ for $j = 1, \ldots, d$. Recall $k^j = |\{l_v : v \in V^j\}|$. Notice that the arc-disjoint s-rooted trees exist in $\mathcal{N}$ form Lemma 8 since $\mathcal{N}$ is fully connected, and hence in order to construct G^j we do not have to compute A^j explicitly and we need only V^j. If we know V^j and l_v and $lev(v)$ of all $v \in V$, we can obtain k^j and $\{L_1^j, \ldots, L_{k^j}^j\}$ in $O(n)$ time for each $j = 1, \ldots, d$. For all $j = 1, \ldots, d$, we can obtain V^j by depth-first search for all $u \in P_s$ in $O(|P_s|m)$ time. In order to add the vertices and arcs to allocate the supplies, we have to compute $V(i, Q)$. First we obtain R_v for all $v \in V$ by depth-first search for all $u \in P_s$ in $O(|P_s|m)$ time. Then, we partition V according to $lev(v)$ in $O(n)$ time to obtain the set of vertices v whose $lev(v)$ takes the same value. Next we assign the value $2^0, 2^1, \ldots, 2^{|P_s|}$ to each $u \in P_s$. Then, for each set of vertices v whose $lev(v)$ takes the same value (say W), we compute the sum of the value of $u \in R_v$ for each $v \in W$, and sort the vertices $v \in W$ by the sum of the value of $u \in R_v$. Notice that for $u, v \in V$ with $R_u \neq R_v$ the sum of the value of the vertices in R_v never be equal to that of the vertices in R_v. The time required to complete this operation for all levels $i = 1, \ldots, k$ is $O(|P_s|n + n \log n)$. From the above discussion, the time required to construct $\mathcal{R}(\mathcal{N})$ is $O(|P_s|m + n \log n)$. $\square$