Solving Evacuation Problems Efficiently
—
Earliest Arrival Flows with Multiple Sources*

Nadine Baumann
Universität Dortmund, FB Mathematik
44221 Dortmund, Germany
nadine.baumann@math.uni-dortmund.de

Martin Skutella
Universität Dortmund, FB Mathematik
44221 Dortmund, Germany
martin.skutella@uni-dortmund.de

Abstract

Earliest arrival flows capture the essence of evacuation planning. Given a network with capacities and transit times on the arcs, a subset of source nodes with supplies and a sink node, the task is to send the given supplies from the sources to the sink "as quickly as possible". The latter requirement is made more precise by the earliest arrival property which requires that the total amount of flow that has arrived at the sink is maximal for all points in time simultaneously.

It is a classical result from the 1970s that, for the special case of a single source node, earliest arrival flows do exist and can be computed by essentially applying the Successive Shortest Path Algorithm for min-cost flow computations. While it has previously been observed that an earliest arrival flow still exists for multiple sources, the problem of computing one efficiently has been open for many years. We present an exact algorithm for this problem whose running time is strongly polynomial in the input plus output size of the problem.

1. Introduction

In typical evacuation situations, the most important task is to get people out of an endangered building or area as fast as possible. Since it is usually not known how long a building can withstand a fire before it collapses or how long a dam can resist a flood before it breaks, it is advisable to organize an evacuation such that as much as possible is saved no matter when the inferno will actually happen. In the more abstract setting of network flows over time, the latter requirement is captured by so-called earliest arrival flows. Before we discuss this in more detail, we first give a short and descriptive introduction into flows over time.

*This work was supported by DFG Focus Program 1126, "Algorithmic Aspects of Large and Complex Networks", grant no. SK 58/4-1 and SK 58/5-3.

Flows over time. We consider a network $N = (V, A)$ with capacities $u_e \geq 0$ and transit times $\tau_e \geq 0$ on the arcs $e \in A$. The capacity of an arc bounds the flow rate (i.e., flow per time) at which flow can enter the arc. The transit time of an arc specifies the amount of time it takes for flow to travel from the tail to the head of the arc. Moreover, there is a set of source nodes $S^+ \subseteq V$ and a set of sink nodes $S^- \subseteq V \setminus S^+$. Each source $s \in S^+$ has a supply $v(s) > 0$ and each sink $t \in S^-$ a demand $-v(t) > 0$ such that

$$\sum_{w \in S^+ \cup S^-} v(w) = 0 \ .$$

A *flow over time*[1] specifies for each arc e and each point in time the flow rate at which flow enters the arc (and leaves the arc again τ_e time units later). Flow conservation constraints require that at every point in time and for every intermediate node $w \in V \setminus (S^+ \cup S^-)$ the flow entering and leaving node w must cancel out each other.

Flows over time have been introduced by Ford and Fulkerson [8] (see also [9]). Given a network with a single source node s, a single sink node t, and a time horizon $\theta \geq 0$, they consider the problem of sending as much flow as possible from s to t within time θ. It turns out that a maximal s-t-flow over time can be determined by a static[2] min-cost flow computation where transit times of arcs are interpreted as cost coefficients.

Ford and Fulkerson [8] also introduce the concept of *time-expanded networks* that consist of one copy of the node set of the given network for each time unit (we call such a copy a *time layer*). For each arc e of the original network with transit time τ_e the time-expanded network contains copies connecting any two time layers at distance τ_e. For more details, we refer to [8, 5]. On the positive side, most flow over time problems can be solved by static flow computations in

[1]There exist two different but closely related models for flows over time—a discrete and a continuous model. We consider the continuous model but the presented results also hold in the discrete model. For more details on this issue we refer to [7].

[2]In order to distinguish them from flows over time, we refer to classical network flows also as *static* flows.

time-expanded networks. On the negative side, time-expanded networks are huge in theory and in practice. In particular, the size of a time expanded network is linear in the given time horizon θ and therefore exponential (but still pseudopolynomial) in the input size.

Hoppe and Tardos [15] consider the *quickest transshipment problem* which is defined as follows. Given a network with several source and sink nodes with given supplies and demands, find a flow over time with minimal time horizon θ that satisfies all supplies and demands. Hoppe and Tardos give a strongly polynomial algorithm for this problem. They present their result for the discrete time model. Fleischer and Tardos [7] show that it also holds in the continuous time model.

Earliest arrival flows. Shortly after Ford and Fulkerson introduced flows over time, the more elaborate *s-t-earliest arrival flow problem* was studied by Gale [10]. Here the goal is to find a single s-t-flow over time that simultaneously maximizes the amount of flow reaching the sink t up to any time $\theta \geq 0$. A flow over time fulfilling this requirement is said to have the *earliest arrival property* and is called *earliest arrival flow*. Gale [10] showed that s-t-earliest arrival flows always exist. Minieka [26] and Wilkinson [31] both gave pseudopolynomial-time algorithms for computing earliest arrival flows based on the Successive Shortest Path Algorithm [20, 16, 2]. Hoppe and Tardos [14] present a fully polynomial time approximation scheme for the earliest arrival flow problem that is based on a clever scaling trick.

In a network with several sources and sinks with given supplies and demands, flows over time having the earliest arrival property do not necessarily exist [4]. We give a simple counterexample with one source and two sinks in Figure 1.

For the case of several sources with given supplies and a single sink, however, earliest arrival flows do always exist [27]. This follows, for example, from the existence of lexicographically maximal flows in time-expanded networks; see, e.g., [26]. We refer to this problem as the *earliest arrival transshipment problem*. Hajek and Ogier [12] give the first polynomial time algorithm for the earliest arrival transshipment problem with zero transit times. Fleischer [4] gives an algorithm with improved running time. Fleischer and Skutella [6] use condensed time-expanded networks to approximate the earliest arrival transshipment problem for the case of arbitrary transit times. They give an FPTAS that approximates the time delay as follows: For every time $\theta \geq 0$ the amount of flow that should have reached the sink in an earliest arrival transshipment by time θ, reaches the sink at latest at time $(1+\varepsilon)\theta$. Tjandra [30] shows how to compute earliest arrival transshipments in networks with time dependent supplies and capacities in time polynomial in the time horizon and the total supply at sources. The resulting running time is thus only pseudopolynomial in the input size.

Earliest arrival flows are motivated by applications related to evacuation. In the context of emergency evacuation from buildings, Berlin [1] and Chalmet et al. [3] study the quickest transshipment problem in networks with multiple sources and a single sink. Jarvis and Ratliff [19][3] show that three different objectives of this optimization problem can be achieved simultaneously: (1) Minimizing the total time needed to send the supplies of all sources to the sink, (2) fulfilling the earliest arrival property, and (3) minimizing the average time for all flow needed to reach the sink. Hamacher and Tufecki [13] study an evacuation problem and propose solutions which further prevents unnecessary movement within a building.

Our contribution. While it has previously been observed that earliest arrival transshipments exist in the general multiple-source single-sink setting, the problem of computing one efficiently has been open. All previous algorithms rely on time expansion of the network into exponentially many time layers. We solve this open problem and present an efficient algorithm which, in particular, does not rely on time expansion.

Using a necessary and sufficient criterion for the feasibility of transshipment over time problems by Klinz [21], we first recursively construct the earliest arrival pattern, that is, the piece-wise linear function that describes the time-dependent maximum flow value. Our algorithm employs submodular function minimization within the parametric search framework of Megiddo [24, 25]. As a by-product, we present a new proof for the existence of earliest arrival flows that does not rely on time expansion. We finally show how to turn the earliest arrival pattern into an earliest arrival flow based on the quickest transshipment algorithm of Hoppe and Tardos [15].

The running time of our algorithm is polynomial in the input size plus the number of breakpoints of the earliest arrival pattern. Since the earliest arrival pattern is more or less explicitly part of the output of the earliest arrival transshipment problem, we can say that the running time of our algorithm is polynomially bounded in the input plus output size.

Outline. In the next section we state a necessary and sufficient criterion for the feasibility of transshipment over time problems and apply it to our setting. In Section 3 we give an in-depth analysis of the structure of the earliest arrival pattern and present a recursive algorithm to compute it. How to compute the actual earliest arrival transshipment out of the pattern is finally shown in Section 4.

[3] Strictly speaking, Jarvis and Ratliff [19] only consider the single-source case but their observation also applies to the more general case with multiple sources.

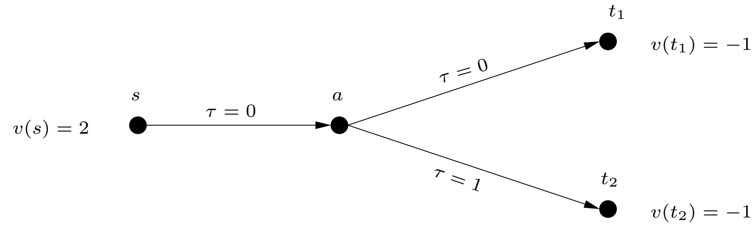

Figure 1. A network with one source and two sinks with unit demands for which an earliest arrival flow does not exist. All arcs have unit capacity and the transit times are given in the drawing. Notice that one unit of flow can reach sink t_1 by time 1; in this case, the second unit of flow reaches sink t_2 only by time 3. Alternatively we can send one unit of flow into sinks t_1 and t_2 simultaneously by time 2. But there does not exist an earliest arrival flow where both flow units must have reached their sink by time 2 and one of them must have already arrived at time 1.

2. Preliminaries

We consider a network with capacities and transit times on the arcs, source nodes S^+ and sink nodes S^- with supplies and demands $v : S^+ \cup S^- \to \mathbb{R}$. We make use of the following result of Klinz [21] (see also [7]).

Lemma 2.1 (Klinz [21]). *For $\theta \geq 0$ and $X \subseteq S^+ \cup S^-$ let*

$$v(X) := \sum_{w \in X} v(w)$$

and let $o^\theta(X)$ be the maximal amount of flow that can be sent from the sources $S^+ \cap X$ to the sinks $S^- \setminus X$ within time θ (ignoring supplies and demands). There exists a flow over time with time horizon θ that satisfies all supplies and demands if and only if

$$o^\theta(X) \;\geq\; v(X) \qquad \text{for all } X \subseteq S^+ \cup S^-.$$

For $\theta \geq 0$ and $X \subseteq S^+ \cup S^-$, the value $o^\theta(X)$ can be obtained by a static min-cost flow computation. Consider the extended network N' defined as follows. Starting from N, introduce a super source s that is connected to all sources $S^+ \cap X$ by an uncapacitated arc with transit time zero and a super sink t that can be reached from all sinks $S^- \setminus X$ by such an arc. By construction of N', the value $o^\theta(X)$ is equal to the value of a maximal s-t-flow over time in N' with time horizon θ. Further extend N' by adding an uncapacitated dummy arc from t to s. It follows from the work of Ford and Fulkerson [8] that

$$o^\theta(X) \;=\; -\min\left\{\mathrm{cost}^\theta(x) \mid x \text{ circulation in } N'\right\} \;. \tag{1}$$

Here, $\mathrm{cost}^\theta(x)$ denotes the cost of circulation x where transit times on arcs are interpreted as cost coefficients and the cost coefficient of dummy arc (t, s) is $-\theta$. As a consequence of (1), the function $\theta \mapsto o^\theta(X)$ is the cost function of a parametric min-cost flow problem. As such, it is piecewise linear and convex.

Based on the work of Megiddo [23], Hoppe and Tardos [15] observe that the function $o^\theta : S^+ \cup S^- \to \mathbb{R}$ is submodular, that is,

$$o^\theta(X) + o^\theta(Y) \;\geq\; o^\theta(X \cup Y) + o^\theta(X \cap Y)$$

for all $X, Y \subseteq S^+ \cup S^-$.
In the following we restrict to networks with a single sink t. The *earliest arrival pattern* $p : \mathbb{R}^+ \to \mathbb{R}^+$ is defined by setting $p(\theta)$ to the maximal amount of flow that can be sent into the sink by time θ without violating supplies at the sources. An *earliest arrival transshipment* is a flow over time such that $p(\theta)$ units of flow have arrived at the sink by time θ for all $\theta \geq 0$ simultaneously.

For the case of a single source $S^+ = \{s\}$ with unbounded supply, the s-t-earliest arrival pattern is $p(\theta) = o^\theta(\{s\})$ and thus piecewise linear and convex. For the case of several sources, the earliest arrival pattern p is still piecewise linear (see Corollary 2.3 below) but not necessarily convex. A simple example with two sources is given in Figure 2. Notice that in this example the rate of flow arriving at the sink (i. e., the derivative of p) suddenly decreases since the entire supply of source s_1 has arrived and this source has therefore run empty. In Section 3 we will observe this effect in a more general context.

The following lemma is essentially a reformulation of Lemma 2.1 for the setting of earliest arrival transshipments and will later turn out to be useful. The proof is technical and will be contained in the full verion of the paper.

Lemma 2.2. *Let $\theta, q \geq 0$. Then $p(\theta) \geq q$ if and only if*

$$o^\theta(S') \;\geq\; q - v(S^+ \setminus S') \quad \text{for all } S' \subseteq S^+. \tag{2}$$

As a consequence of Lemma 2.2, we can show that the earliest arrival pattern is a piecewise linear function.

Corollary 2.3. *The earliest arrival pattern p is piecewise linear.*

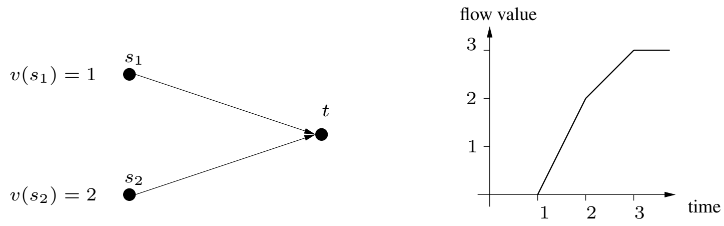

Figure 2. A simple example of a graph with two sources, unit capacities, and unit transit times where the optimal arrival pattern of a feasible earliest arrival transshipment is piecewise linear and non-convex.

Proof. As a result of Lemma 2.2 we get

$$p(\theta) \;=\; \min\{o^\theta(S') + v(S^+ \setminus S') \mid S' \subseteq S^+\} \enspace .$$

Since $\theta \mapsto o^\theta(S')$ is a piecewise linear (and convex) function for all $S' \subseteq S^+$, the result follows. $\square$

In the next section we show how we can determine the earliest arrival pattern of the earliest arrival transshipment problem. The earliest arrival transshipment itself can then be obtained from the given earliest arrival pattern as shown in Section 4.

3. Constructing the earliest arrival pattern

Throughout this section we use the following example instance to illustrate the presented ideas and techniques.

Example. *Assume we are given a network as on the left hand side in Figure 3 with unit transit times and unit capacities. The supplies of the sources are as given in the picture.*

3.1. The structure of the earliest arrival pattern

We show that the earliest arrival pattern p is composed of several s-t-earliest arrival patterns in extended networks with an additional supersource s that is connected to certain subsets of sources in S^+. We start by considering the extended network N_0 that arises from connecting supersource s to all nodes in S^+ by an uncapacitated, zero transit time arc. The nodes in S^+ are no longer sources but take the role of intermediate nodes in N_0 and their total supply $v(S^+)$ is shifted to the supersource s. Thus, a feasible s-t-flow over time in the extended network N_0 induces a flow over time in N where $v(S^+)$ units of flow are being sent from the sources in S^+ to sink t. Notice, however, that the induced flow over time in N might violate individual supplies at the source nodes.

The s-t-earliest arrival pattern in N_0 is the function $\theta \mapsto o^\theta(S^+)$. For every $\theta \geq 0$ it holds that $p(\theta) \leq o^\theta(S^+)$. If $p(\theta) = o^\theta(S^+)$ for all $\theta \geq 0$, we are done since we know how to obtain the s-t-earliest arrival pattern $\theta \mapsto o^\theta(S^+)$. Otherwise, let $\theta_1 := \sup\{\theta \mid p(\theta) = o^\theta(S^+)\}$.[4] We use the following lemma to prove that $p(\theta) = o^\theta(S^+)$ for all $0 \leq \theta \leq \theta_1$.

Lemma 3.1. *Let $S'' \subseteq S' \subseteq S^+$ and $0 \leq \theta' \leq \theta$. Then,*

$$o^{\theta'}(S') - o^{\theta'}(S'') \;\leq\; o^\theta(S') - o^\theta(S'') \enspace .$$

Proof. Consider an extended network $\bar{N}$ with an additional sink t' that can be reached from t through an uncapacitated arc (t, t') with transit time $\theta - \theta'$. The underlying intuition is that all flow arriving at t before time θ' can be forwarded to the new sink t' where it arrives before time θ. For $\bar{S} \subseteq S^+ \cup \{t, t'\}$ let $\bar{o}^\theta(\bar{S})$ denote the maximum amount of flow that can be sent from the sources in $\bar{S}$ to the sinks in $(S^+ \cup \{t, t'\}) \setminus \bar{S}$ by time θ. By construction of $\bar{N}$ we get for $\bar{S} \subseteq S^+$ the following equalities:

$$\bar{o}^\theta(\bar{S}) \;=\; o^\theta(\bar{S}) \quad \text{and} \quad \bar{o}^\theta(\bar{S} \cup \{t\}) \;=\; o^{\theta'}(\bar{S}) \enspace . \tag{3}$$

We can now prove the statement of the lemma. By (3) and submodularity of $\bar{o}^\theta(\cdot)$ we get

$$\begin{aligned} o^{\theta'}(S') - o^{\theta'}(S'') \;&=\; \bar{o}^\theta(S' \cup \{t\}) - \bar{o}^\theta(S'' \cup \{t\}) \\ &\leq\; \bar{o}^\theta(S') - \bar{o}^\theta(S'') \\ &=\; o^\theta(S') - o^\theta(S'') \enspace . \end{aligned}$$

This concludes the proof. $\square$

Corollary 3.2. *Let $\theta_1 = \max\{\theta \mid p(\theta) = o^\theta(S^+)\}$. Then $p(\theta) = o^\theta(S^+)$ for all $0 \leq \theta \leq \theta_1$.*

Proof. Assume by contradiction that $p(\theta) < o^\theta(S^+)$ for some $0 \leq \theta < \theta_1$. By Lemma 2.2 there exists $S' \subseteq S^+$ with

$$o^\theta(S') \;<\; o^\theta(S^+) - v(S^+ \setminus S') \enspace .$$

It follows from Lemma 3.1 that

$$o^{\theta_1}(S') \;<\; o^{\theta_1}(S^+) - v(S^+ \setminus S')$$

such that $p(\theta_1) < o^{\theta_1}(S^+)$ by Lemma 2.2. This contradicts the choice of θ_1. $\square$

[4] The supremum here is indeed a maximum since $p(\theta)$ and $o^\theta(S^+)$ are both continuous functions of θ.

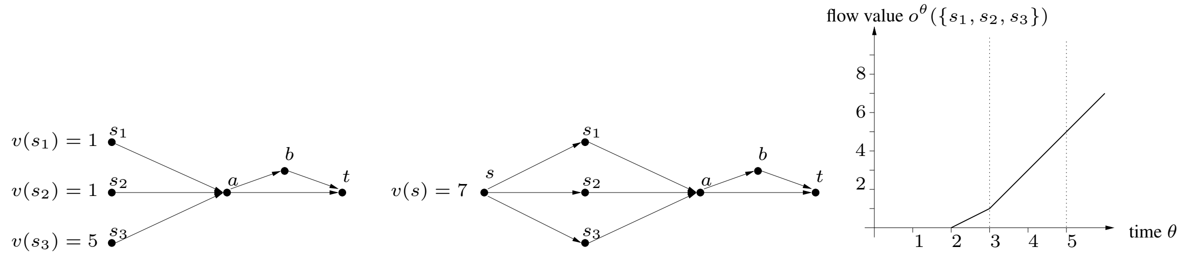

Figure 3. Example of a network $N = (V, A)$, the network expanded by a supersource, and the corresponding s-t-earliest arrival pattern.

Example. *In order to compute the s-t-earliest arrival pattern for the network given in the left part of Figure 3 we insert a supersource s as depicted in the middle part of Figure 3. Applying the Successive Shortest Path Algorithm to this network yields, for example, the two paths $P_1 = (s, s_1, a, t)$ and $P_2 = (s, s_3, a, b, t)$, both with flow rate 1. The resulting arrival pattern up to time 6 is given in the right part of Figure 3.*

Notice that the flow arriving at sink node t after time 3 violates the supply of node s_1 since more than one unit of flow has been sent through path P_1. On the other hand it can easily be seen that we can reroute the flow gaining a path decomposition with $P_1' = (s, s_3, a, t)$, $P_2' = (s, s_1, a, b, t)$, and $P_3' = (s, s_2, a, b, t)$ where the flow rate on path P_1' is 1 and the flow rates on paths P_2' and P_3' are only $1/2$. Notice that the flow arriving over these paths at the sink does not violate supplies up to time 5 and has still the same arrival pattern. Further, there is no other way of sending flow obeying the supplies of sources s_1, s_2, s_3 for longer than 5 time units. After time 5 the slope of the earliest arrival pattern p decreases since no more flow out of sources s_1 and s_2 can reach the sink. In particular, the value of θ_1 equals 5.

In our example, any flow over time in N that sends $p(\theta_1)$ units into the sink t by time θ_1 must use up the supplies of sources s_1 and s_2. In other words, the bounded supplies of these sources are the reason why $p(\theta) < o^\theta(S^+)$ for $\theta > \theta_1$. The next lemma illuminates this effect for general instances.

Lemma 3.3. *There exists a subset of sources $S_1 \subsetneq S^+$ such that*

$$o^{\theta_1}(S_1) \;=\; o^{\theta_1}(S^+) - v(S^+ \setminus S_1) \;\; .$$

Before we prove the lemma, we first give an intuitive interpretation of its statement. In an earliest arrival transshipment, $p(\theta_1) = o^{\theta_1}(S^+)$ units of flow reach the sink by time θ_1. The lemma states that at most $o^{\theta_1}(S^+) - v(S^+ \setminus S_1)$ of these units can originate from sources in S_1. The remaining $v(S^+ \setminus S_1)$ units must originate from sources in $S^+ \setminus S_1$. These sources therefore run empty and cannot contribute to flow arriving after time θ_1 at the sink.

Proof. By contradiction assume that

$$o^{\theta_1}(S') \;>\; o^{\theta_1}(S^+) - v(S^+ \setminus S') \quad \text{for all } S' \subsetneq S^+.$$

Since $o^\theta(S')$ and $o^\theta(S^+)$ are continuous functions of θ, there exists $\epsilon > 0$ such that

$$o^{\theta_1+\epsilon}(S') \;\geq\; o^{\theta_1+\epsilon}(S^+) - v(S^+ \setminus S')$$

for all $S' \subseteq S^+$. By Lemma 2.2 this implies

$$p(\theta_1 + \epsilon) \geq o^{\theta_1+\epsilon}(S^+) \;\; .$$

This contradicts the choice of θ_1. □

We consider the reduced instance of the earliest arrival transshipment problem that is obtained by setting the supplies of all sources in $S^+ \setminus S_1$ to zero. The earliest arrival pattern of the modified instance is denoted by p'. The following theorem is the main result of this section.

Theorem 3.4. *Let $\theta_1 = \max\{\theta \mid p(\theta) = o^\theta(S^+)\}$ and $S_1 \subsetneq S^+$ such that*

$$o^{\theta_1}(S_1) = o^{\theta_1}(S^+) - v(S^+ \setminus S_1)$$

(see Lemma 3.3). Let p' denote the earliest arrival pattern of the modified instance with source set S_1. Then,

$$p(\theta) \;=\; \begin{cases} o^\theta(S^+) & \text{if } \theta < \theta_1, \\ p'(\theta) + v(S^+ \setminus S_1) & \text{if } \theta \geq \theta_1. \end{cases}$$

As a result of Theorem 3.4, we have reduced the problem of constructing the earliest arrival pattern p to the problem of computing an s-t-earliest arrival pattern and computing an earliest arrival pattern for a smaller number of sources S_1.

Proof. It follows from Corollary 3.2 that $p(\theta) = o^\theta(S^+)$ for $\theta \leq \theta_1$. It remains to show that

$$p(\theta) \;=\; p'(\theta) + v(S^+ \setminus S_1) \qquad \text{for all } \theta \geq \theta_1.$$

It is clear that "$\leq$" holds since by time θ at most $p'(\theta)$ and $v(S^+ \setminus S_1)$ units of flow can reach the sink originating from sources in S_1 and $S^+ \setminus S_1$, respectively.

It remains to show that "$\geq$" holds, that is, $p'(\theta) + v(S^+ \setminus S_1)$ units of flow can be sent into the sink t by time $\theta \geq \theta_1$ without exceeding supplies at the sources. We check the condition given in Lemma 2.2. For $S' \subseteq S^+$ and $\theta \geq \theta_1$ we get by submodularity of $o^\theta(\cdot)$:

$$o^\theta(S') \;\geq\; o^\theta(S' \cap S_1) + o^\theta(S' \cup S_1) - o^\theta(S_1)$$

by Lemma 3.1:

$$\geq\; o^\theta(S' \cap S_1) + o^{\theta_1}(S' \cup S_1) - o^{\theta_1}(S_1)$$

by Lemma 2.2 and Lemma 3.3:

$$\begin{aligned}
&\geq\; \big(p'(\theta) - v(S_1 \setminus S')\big) \\
&\qquad + \Big(o^{\theta_1}(S^+) - v\big(S^+ \setminus (S' \cup S_1)\big)\Big) \\
&\qquad - \Big(o^{\theta_1}(S^+) - v(S^+ \setminus S_1)\Big) \\
&=\; p'(\theta) - v(S_1 \setminus S') - v\big(S^+ \setminus (S' \cup S_1)\big) \\
&\qquad + v(S^+ \setminus S_1) \\
&=\; p'(\theta) - v(S^+ \setminus S') + v(S^+ \setminus S_1) \;\; .
\end{aligned}$$

The result now follows from Lemma 2.2. $\square$

Example. *For our example given in Figure 3 we have already seen that up to time $\theta_1 = 5$ flow of value 5 including the total supply of the sources s_1 and s_2 can be sent into the sink. In particular, it holds that*

$$o^\theta(S') \geq o^\theta(S^+) - v(S^+ \setminus S')$$

for all $S' \subseteq S^+$ and $\theta \leq \theta_1$. For the set $S_1 := \{s_3\} \subseteq S^+$ and $\theta = \theta_1$ this inequality is tight. The function $\theta \mapsto o^\theta(S^+)$ is already known (see the right part of Figure 3).

For the restricted earliest arrival problem with sources $S_1 = \{s_3\}$, the earliest arrival pattern p' is given in the left part of Figure 4. By Theorem 3.4, the resulting earliest arrival pattern p of the original instance is the lower envelop of the two functions depicted in the right part of Figure 4.

3.2. Computing the earliest arrival pattern

With Theorem 3.4 we have reduced the problem of computing the earliest arrival pattern to an s-t-earliest arrival flow problem and an earliest arrival transshipment problem on a reduced instance with a strictly smaller set of sources. Applying this result recursively to the reduced instance finally yields Algorithm 1 which computes the earliest arrival pattern p.

For the understanding of the algorithm it is helpful to observe that $\theta_i < \theta_{i+1}$ for all $i \geq 0$. The statement is clear for $i = 0$ since the sources in $S^+ \setminus S_1$

Algorithm 1: Computing the earliest arrival pattern.

Input: (G, S^+, t)
Output: Earliest arrival pattern p.

1 set $i := 0$, $S_i := S^+$, and $\theta_i := 0$;
2 **while** $S_i \neq \emptyset$ **do**
3 compute the maximal value $\theta_{i+1} \geq 0$ such that $o^{\theta_{i+1}}(S') \;\geq\; o^{\theta_{i+1}}(S_i) - v(S_i \setminus S')$ for all $S' \subseteq S_i$;
4 compute an inclusion-wise minimal $S_{i+1} \subsetneq S_i$ with
$$o^{\theta_{i+1}}(S_{i+1}) \;=\; o^{\theta_{i+1}}(S_i) - v(S_i \setminus S_{i+1}) \;; \tag{4}$$
5 compute the function $\theta \mapsto o^\theta(S_i)$ on the interval $[\theta_i, \theta_{i+1})$ and set $p(\theta) := o^\theta(S_i) + v(S^+ \setminus S_i)$ for $\theta \in [\theta_i, \theta_{i+1})$;
6 $i := i + 1$;
7 set $p(\theta) := v(S^+)$ for all $\theta \geq \theta_i$;

have positive supply and therefore cannot run empty at time $\theta_0 = 0$. For $i \geq 1$ assume by contradiction that $\theta_{i+1} \leq \theta_i$. This yields by Lemma 3.1:

$$o^{\theta_i}(S_{i+1}) \;\leq\; o^{\theta_i}(S_i) + o^{\theta_{i+1}}(S_{i+1}) - o^{\theta_{i+1}}(S_i)$$

by (4):

$$=\; o^{\theta_i}(S_i) - v(S_i \setminus S_{i+1})$$

by (4) with $i := i - 1$:

$$\begin{aligned}
&=\; o^{\theta_i}(S_{i-1}) - v(S_{i-1} \setminus S_i) \\
&\qquad - v(S_i \setminus S_{i+1}) \\
&=\; o^{\theta_i}(S_{i-1}) - v(S_{i-1} \setminus S_{i+1})
\end{aligned}$$

which contradicts the minimal choice of $S_i \supsetneq S_{i+1}$ in step 4 of the algorithm.

Theorem 3.5. *Algorithm 1 computes the earliest arrival pattern and can be implemented to run in strongly polynomial time in the input plus output size.*

In order to prove this theorem, we need the following technical lemma which gives a bound on the computational complexity of step 5.

Lemma 3.6. *For $0 \leq \theta_i \leq \theta_{i+1}$ and $S' \subseteq S^+$, the piecewise linear function $g : [\theta_i, \theta_{i+1}) \to \mathbb{R}$ with $g(\theta) := o^\theta(S')$ can be computed in time polynomial in the input size plus the number of breakpoints.*

Proof. In order to compute $g(\theta) = o^\theta(S')$, we consider the extended network N' that is obtained as follows. Add a supersource s that is connected to all sources in S' by an uncapacitated arc with transit time zero and that can be reached from t by an uncapacitated dummy arc (t, s). As already stated in (1), $g(\theta)$ is equal to the cost of a min-cost circulation in N'

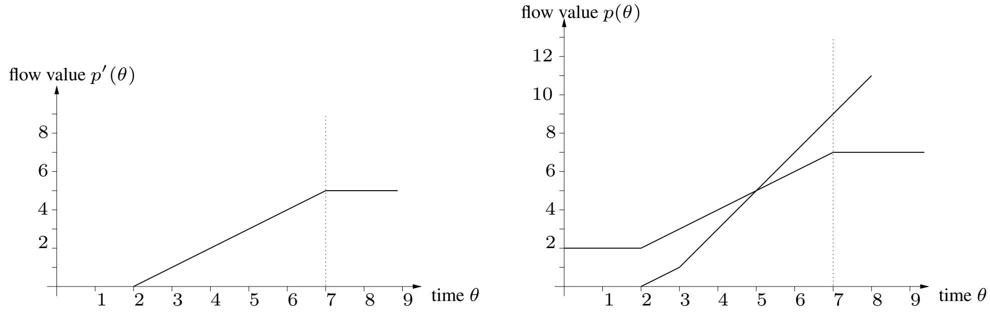

Figure 4. Optimal pattern p' for the problem with the reduced set of sources S_1 (left) and the combined pattern p as the lower bound of the line segments (right).

where the cost coefficient of the dummy arc (t, s) is set to $\tau_{(t,s)} = -\theta$. We denote the cost of an arbitrary circulation x in this network by $\text{cost}^\theta(x)$.

We start by computing a min-cost circulation x in N' for $\theta = \theta_i$. Let N'_x denote the residual network of x and let θ' be the length of a shortest s-t-path in N'_x. Since there is the uncapacitated dummy arc (t, s) of cost $-\theta_i$ in N'_x, optimality of x implies $\theta' \geq \theta_i$. Moreover, for all $\theta \in [\theta_i, \theta']$, the circulation x is still a min-cost circulation and $g(\theta) = \text{cost}^\theta(x)$. Since the cost of x depends linearly on θ, the function g is thus linear on the interval $[\theta_i, \theta']$. If $\theta' \geq \theta_{i+1}$, then we are done. Otherwise we have discovered a breakpoint of g at θ'. Notice that x is no longer optimal for $\theta > \theta'$ since the cost can be reduced by augmenting flow on a negative cycle formed by a shortest s-t-path of length θ' in N'_x and the dummy arc (t, s) of length $-\theta$.

We obtain the next linear piece of g starting at θ' as follows. Compute the subnetwork N''_x of the residual network N'_x that is formed by all arcs that lie on some shortest s-t-path. Compute a maximum s-t-flow in N''_x and turn it into a circulation y in N'_x by sending all flow from t back to s on the dummy arc (t, s). Augmenting x according to y yields a new circulation x. The new circulation is optimal for all $\theta \in [\theta', \theta'']$ where $\theta'' > \theta'$ is the length of a shortest s-t-path in the new residual network N'_x and determines the next breakpoint of g.

The described process is iterated until the length of a shortest s-t-path in the residual network is at least θ_{i+1}. Notice that the overall running time is dominated by the initial min-cost flow computation plus number of breakpoints many max-flow computations. $\square$

Example. *In our example depicted in Figure 3 we can find the function $g : [\theta_i, \theta_{i+1}) \to \mathbb{R}$ as described above. For the interval $[\theta_0, \theta_1)$ we get the networks N', N'_x, and N''_x as follows. Network N' is constructed by adding a supersource s connected to all sources by uncapacitated, zero transit time arcs and an uncapacitated arc (t, s) with transit time $\tau_{(t,s)} = -2$. This is depicted in Figure 5.1. In this network we compute a min-cost (maximum flow) circulation by sending one unit of flow for example over cycle s, s_2, a, t, s. This yields the residual network N'_x which is depicted in Figure 5.2. There the shortest s-t-path, for example path s, s_3, a, b, t, has length $\theta' = 3$. In the subnetwork N''_x consisting of all arcs being part of some shortest s-t-path we now compute a maximum s-t-flow. Such a path flow s, s_1, a, b, t is depicted in Figure 5.3. Reconsidering network N'_x together with a new circulation of one unit of flow along cycle s, s_1, a, b, t, s results in the new residual network N'_x(new) which is depicted in Figure 5.4. There no (shortest) s-t-path remains and therefore the transit time of arc (t, s) is set to infinity which is strictly greater than θ_1. Thus we have found function g on the interval $[\theta_0, \theta_1)$ which is of the form shown in Figure 5.5.*

Proof of Theorem 3.5. The correctness of the algorithm follows from Section 3.1 and in particular from Theorem 3.4. It thus remains to prove the stated bound on the running time.

First notice that the number of iterations of the while-loop in step 2 is bounded by the number of sources since at least one source is eliminated from S_i in every iteration. Since step 5 can be done in strongly polynomial time, it remains to show that steps 3 and 4 can also be done in strongly polynomial time.

We start with the computation of θ_{i+1} in step 3. For $\theta \geq 0$ we define the function $f^\theta : 2^{S_i} \to \mathbb{R}$ by

$$f^\theta(S') := o^\theta(S') - o^\theta(S_i) + v(S_i \setminus S')$$

for $S' \subseteq S_i$. Computing θ_{i+1} thus amounts to finding the maximal value $\theta \geq 0$ such that

$$f^\theta(S') \geq 0 \qquad \text{for all } S' \subseteq S_i. \tag{5}$$

Since o^θ is submodular and the function

$$S' \mapsto v(S_i \setminus S') - o^\theta(S_i)$$

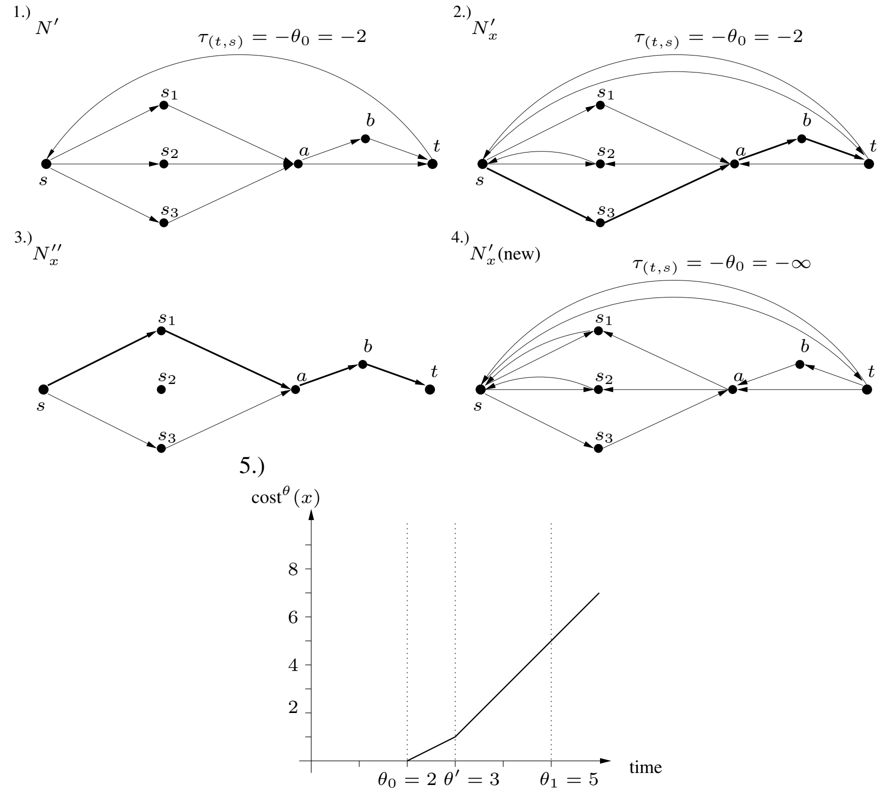

Figure 5. Networks used to compute the function $g : [\theta_0, \theta_1) \to \mathbb{R}$ as described in the proof of Lemma 3.6 for the example given in Figure 3.

is modular, f^θ is submodular. According to (1), computing $f^\theta(S')$ for some $S' \subseteq S_i$ requires two min-cost flow computations where the cost coefficients depend linearly on the parameter θ. It was shown by Grötschel, Lovász, and Schrijver [11] that there is a strongly polynomial algorithm for minimizing a submodular function[5]. It can therefore be tested in strongly polynomial time whether (5) is fulfilled for a fixed value θ. Embedding this algorithm into Megiddo's parametric search framework (see [24, 25]) gives a procedure for step 3 whose running time is strongly polynomial in the input size of our problem (more details can be found in [15]).

We finally discuss how to compute S_{i+1} in step 4 in strongly polynomial time. Notice that (4) translates to $f^{\theta_{i+1}}(S_{i+1}) = 0$, that is, S_{i+1} minimizes the submodular function $f^{\theta_{i+1}}$. By submodularity of $f^{\theta_{i+1}}$, there exists a unique inclusion-wise minimal subset S_{i+1} which can be obtained as follows[6] (see, e.g., [29, Chapter 45]). Initialize $S_{i+1} := S_i$. For each $s \in S_i$, check whether the minimum value of $f^{\theta_{i+1}}$ over all subsets of $S_{i+1} \setminus \{s\}$ is zero. If so, reset $S_{i+1} := S_{i+1} \setminus \{s\}$. Doing this for all elements of S_i finally yields the unique inclusion-wise minimal subset S_{i+1} with $f^{\theta_{i+1}}(S_{i+1}) = 0$. A faster algorithm for computing the inclusion-wise minimal subset S_{i+1} was recently given by McCormick and Queyranne [22]. □

[5]Combinatorial algorithms achieving strongly polynomial running time are given by Iwata, Fleischer, and Fujishige [18] and by Schrijver [28]. A fully combinatorial algorithm is given by Iwata [17].

[6]For the purpose of our algorithm it is of course advantageous to choose the minimal subset S_{i+1} in order to reduce the number of sources as far as possible.

4. Turning the earliest arrival pattern into an earliest arrival transshipment

In this section we assume that we are given the piecewise linear earliest arrival pattern p of the earliest arrival transshipment problem by its breakpoints $(x_0, f_0), (x_1, f_1), \ldots, (x_k, f_k)$, that is,

$$p(\theta) = \begin{cases} 0 & \text{if } \theta \leq x_0, \\ f_i + \frac{\theta - x_i}{x_{i+1} - x_i}(f_{i+1} - f_i) & \text{if } x_i \leq \theta, \ \theta \leq x_{i+1}, \ 0 \leq i < k, \\ f_k & \text{if } \theta \geq x_k. \end{cases}$$

An illustration is given in Figure 6. Notice that the values x_i determine points in time and the values f_i determine an amount of flow for all i.

Further notice that

$$x_0 < x_1 < \cdots < x_k$$

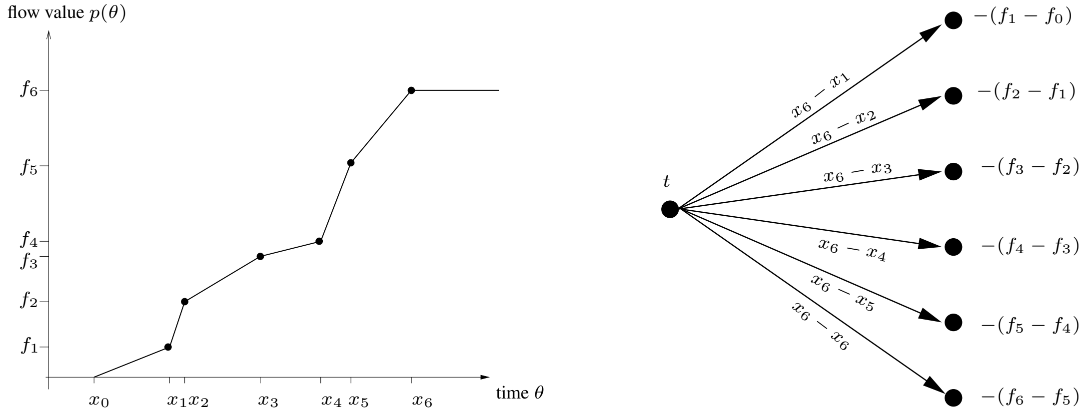

Figure 6. On the left hand side we draw the earliest arrival pattern p with breakpoints (x_i, f_i), $i = 1, 2, \dots, k = 6$. On the right hand side the modified network is depicted. The capacity of arc $e_i = (t, t_i)$ is set to $(f_i - f_{i-1})(x_i - x_{i-1})$.

and x_0 is the first point in time when flow can reach the sink (i. e., x_0 is the transit time of a shortest path leading from any source to the sink). Moreover,

$$0 = f_0 \leq f_1 \leq \cdots \leq f_k = v(S^+) \ .$$

We show that the problem of finding an earliest arrival transshipment can be reduced to finding a transshipment over time in a slightly modified network N' with k additional arcs leading from t to k new sink nodes $t_1, \dots, t_k$. An illustration of the modification is given in Figure 6.

Node t is no longer a sink but just an intermediate node of the modified network N'. For $i = 1, \dots, k$, the demand of sink t_i is set to $-(f_i - f_{i-1})$ such that the total demand $-f_k$ of the sinks and the total supply $v(S^+)$ at the sources cancel out each other. The arc leading from t to sink t_i is called e_i. The transit time of arc e_i is defined to be $\tau_{e_i} := x_k - x_i$, its capacity is $(f_i - f_{i-1})/(x_i - x_{i-1})$ and thus equal to the derivative of p within the interval $[x_{i-1}, x_i]$. Notice that the capacity of e_i is chosen such that the demand of sink t_i is fulfilled if flow is being sent at maximal rate into arc e_i within time interval $[x_{i-1}, x_i)$. As a consequence of this observation, we can state the following lemma.

Lemma 4.1. *An earliest arrival flow in N with earliest arrival pattern p naturally induces a feasible transshipment over time with time horizon x_k satisfying all supplies and demands in N'*

Proof. Take an earliest arrival flow in N and turn it into a transshipment over time in N' by sending all flow arriving at t in time interval $[x_{i-1}, x_i)$ to t_i along arc e_i. □

The reverse direction of Lemma 4.1 also holds. Due to space limitations, we omit this proof in this extended abstract. It will be contained in the full version of the paper.

Lemma 4.2. *A transshipment over time with time horizon x_k that satisfies all supplies and demands in the modified network N' naturally induces an earliest arrival transshipment in N.*

We finally prove that a transshipment over time with time horizon x_k that satisfies all supplies and demands in the modified network N' actually exists. As a consequence of Lemma 4.2, this yields a new proof for the existence of an earliest arrival transshipment in N.

Lemma 4.3. *There exists a transshipment over time with time horizon x_k satisfying all supplies and demands in N'.*

Proof. We denote the set of sources in N' by S^+ and the set of sinks by $S^- = \{t_1, \dots, t_k\}$. For an arbitrary $S' \subseteq S^+ \cup S^-$ let $\bar{o}^\theta(S')$ denote the maximum amount of flow that can be sent within time θ from sources $S^+ \cap S'$ to sinks $S^- \setminus S'$. By Lemma 2.1 we have to show that $\bar{o}^\theta(S') \geq v(S')$ for $\theta = x_k$.

Let $o^\theta(S^+ \cap S')$ denote the maximum amount of flow that can be sent within time θ from sources $S^+ \cap S'$ to t. By Lemma 3.1 we get

$$o^\theta(S^+ \cap S') + v(S^+ \setminus S') \geq p(\theta) \quad \text{for all } \theta \geq 0.$$

This inequality can be interpreted as follows: If we assume that the total supply $v(S^+ \setminus S')$ of the sources $S^+ \setminus S'$ is already in t from time zero on, then we can send $v(S^+ \cap S')$ additional flow units from the sources in $S^+ \cap S'$ (ignoring their individual supplies) into t such that the amount of flow at t is at least $p(\theta)$ at any time $\theta \geq 0$. By forwarding flow from t to the sinks in S^- (similar to the proof of Lemma 4.1), we get a flow over time with time horizon x_k that satisfies the demands of all sinks in S^-. From this flow over time we now remove the $v(S^+ \setminus S')$ flow units that we assumed to be in t at time zero. This yields a flow over time with time horizon x_k from the sources in $S^+ \cap S'$

to the sinks S^- such that the total amount of flow sent is $v(S^+ \cap S')$ and no sink in S^- gets more than its demand. Therefore the flow arriving at sinks in $S^- \setminus S'$ is at least $v(S^+ \cap S') + v(S^- \cap S') = v(S')$. We have thus shown that $\bar{o}^\theta(S') \geq v(S')$ for $\theta = x_k$. This concludes the proof. ☐

As a consequence we can state the following theorem.

Theorem 4.4. *Given the earliest arrival pattern p with k breakpoints for network N, an earliest arrival transshipment in N can be obtained by computing a transshipment over time in a modified network N' with k additional nodes and arcs.*

In order to compute a transshipment over time in the modified network N' we can use the algorithm of Hoppe and Tardos [15]. Since the running time of this algorithm is bounded by a polynomial in the encoding size of the input N' and since the encoding size of N' is of the same order as the encoding size of N plus the encoding size of p, the required running time is polynomial in the input plus output size of the earliest arrival flow problem on N.

Acknowledgements. The authors want to thank Lisa Fleischer, Bettina Klinz, and Ekkehard Köhler for interesting and helpful discussions on the topic of this paper.

References

[1] G. N. Berlin. *The use of directed routes for assessing escape potential*. National Fire Protection Association, Boston, MA, 1979.

[2] R. G. Busaker and P. J. Gowen. A procedure for determining minimal-cost network flow patterns. Technical Report 15, Operational Research Office, John Hopkins University, Baltimore, MD, 1961.

[3] L. G. Chalmet, R. L. Francis, and P. B. Saunders. Network models for building evacuation. *Management Science*, 28:86–105, 1982.

[4] L. K. Fleischer. Faster algorithms for the quickest transshipment problem. *SIAM Journal on Optimization*, 12:18–35, 2001.

[5] L. K. Fleischer and M. Skutella. Quickest flows over time. *SIAM Journal on Computing*. To appear.

[6] L. K. Fleischer and M. Skutella. The quickest multicommodity flow problem. In W. J. Cook and A. S. Schulz, editors, *Integer Programming and Combinatorial Optimization*, volume 2337 of *Lecture Notes in Computer Science*, pages 36–53. Springer, Berlin, 2002.

[7] L. K. Fleischer and É. Tardos. Efficient continuous-time dynamic network flow algorithms. *Operations Research Letters*, 23:71–80, 1998.

[8] L. R. Ford and D. R. Fulkerson. Constructing maximal dynamic flows from static flows. *Operations Research*, 6:419–433, 1958.

[9] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, Princeton, NJ, 1962.

[10] D. Gale. Transient flows in networks. *Michigan Mathematical Journal*, 6:59–63, 1959.

[11] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, Berlin, 1988.

[12] B. Hajek and R. G. Ogier. Optimal dynamic routing in communication networks with continuous traffic. *Networks*, 14:457–487, 1984.

[13] H. W. Hamacher and S. Tifecki. On the use of lexicographic min cost flows in evacuation modeling. *Naval Research Logistics*, 34:487–503, 1987.

[14] B. Hoppe and É. Tardos. Polynomial time algorithms for some evacuation problems. In *Proceedings of the 5th Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 433–441, Arlington, VA, 1994.

[15] B. Hoppe and É. Tardos. The quickest transshipment problem. *Mathematics of Operations Research*, 25:36–62, 2000.

[16] M. Iri. A new method of solving transportation-network problems. *Journal of the Operations Research Society of Japan*, 26:27–87, 1960.

[17] S. Iwata. A fully combinatorial algorithm for submodular function minimization. *Journal of Combinatorial Theory, Ser. B*, 84:203–212, 2002.

[18] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48:761–777, 2001.

[19] J. Jarvis and H. Ratliff. Some equivalent objectives for dynamic network flow problems. *Management Science*, 28:106–108, 1982.

[20] P. A. Jewel. Optimal flow through networks. Technical Report 8, Operations Research Center, MIT, Cambridge, MA, 1958.

[21] B. Klinz. Cited as personal communication (1994) in [15].

[22] S. T. McCormick and M. Queyranne. Finding all optimal solutions for submodular function minimization. Talk given at Oberwolfach workshop ID 0545 on Combinatorial Optimization, November 2005.

[23] N. Megiddo. Optimal flows in networks with multiple sources and sinks. *Mathematical Programming*, 7:97–107, 1974.

[24] N. Megiddo. Combinatorial optimization with rational objective functions. *Mathematics of Operations Research*, 4:414–424, 1979.

[25] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *Journal of the ACM*, 30:852–865, 1983.

[26] E. Minieka. Maximal, lexicographic, and dynamic network flows. *Operations Research*, 21:517–527, 1973.

[27] D. Richardson and É. Tardos. Cited as personal communication (2002) in [5].

[28] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80:346–355, 2000.

[29] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer, Berlin, 2003.

[30] S. Tjandra. *Dynamic Network Optimization with Application to the Evacuation Problem*. PhD thesis, Univeristät Kaiserslautern, Shaker Verlag, Aachen, 2003.

[31] W. L. Wilkinson. An algorithm for universal maximal dynamic flows in a network. *Operations Research*, 19:1602–1612, 1971.