

Earliest Arrival Flows with Multiple Sources

Nadine Baumann

TU Dortmund, Fakultät für Mathematik, 44221 Dortmund, Germany

email: nadine.baumann@udo.edu

Martin Skutella

TU Berlin, Institut für Mathematik, Str. des 17. Juni 136, 10623 Berlin, Germany

email: skutella@math.tu-berlin.de <http://www.math.tu-berlin.de/~skutella/>

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.

Key words: network flow; flow over time; earliest arrival flow; evacuation problem

MSC2000 Subject Classification: Primary: 90C27, 90B10, 68Q25; Secondary: 90B06, 90B20

OR/MS subject classification: Primary: networks/graphs, flow algorithms; Secondary: transportation

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.

Flows Over Time. We consider a network $\mathcal{N} = (V, A)$ 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 $t \in S^-$ has a demand $-v(t) > 0$ such that

$$v(S^+ \cup S^-) = 0,$$

where $v(S') := \sum_{w \in S'} v(w)$ for $S' \subseteq S^+ \cup S^-$.

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 rate of flow entering w equals the rate of flow leaving w .

Flows over time have been introduced by Ford and Fulkerson [9] (see also [10]). 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 . They show that a maximal s - t -flow over time can be determined by a static² min-cost flow computation where transit times of arcs are interpreted as cost coefficients.

Ford and Fulkerson [9] 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 θ and θ' with $\theta' - \theta = \tau_e$. For more details, we refer to [9, 7, 35]. On the positive side,

¹There 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 [8].

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

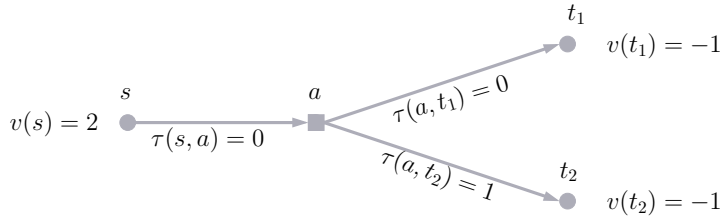


Figure 1: A network with one source and two sinks with unit demands. All arcs have unit capacity and the transit times are given in the drawing. The two flow units at the source can satisfy the demands at the sinks by time 2: First, the flow unit for sink t_2 is pumped into arc (s, a) uniformly during time interval $[0, 1)$; it proceeds along arc (a, t_2) and arrives at t_2 during time interval $[1, 2)$. Then, the flow for sink t_1 is sent into arc (s, a) uniformly during time interval $[1, 2)$; it proceeds along arc (a, t_1) and arrives at t_1 during time interval $[1, 2)$ as well. The described schedule is the only one that satisfies both demands by time 2. On the other hand, if we want to maximize the amount of flow that has reached the sinks by time 1, we have to deviate from this schedule and satisfy the demand at sink t_1 first. Thus, an earliest arrival flow does not exist.

most flow over time problems can be solved by static flow computations in 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 [17] 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 which employs submodular function minimization within the parametric search framework of Megiddo [26, 27]. They present their result for the discrete time model. Fleischer and Tardos [8] 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 *earliest arrival s - t -flow problem* was studied by Gale [11]. 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 s - t -flow*. If we consider flows from more than one source, we omit the prefix “ s - t ”. Gale [11] showed that earliest arrival s - t -flows always exist. Miniéka [28] and Wilkinson [37] both gave pseudopolynomial-time algorithms for computing earliest arrival s - t -flows based on the Successive Shortest Path Algorithm [23, 18, 3]. Hoppe and Tardos [16] present a fully polynomial-time approximation scheme for the earliest arrival s - t -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 [5]. We give a simple counterexample with one source and two sinks in Figure 1. An open problem in this context is being discussed in the concluding section of this paper. Note that we consider flows in networks with given supplies and demands as transshipments in the following.

For the case of several sources with given supplies and a single sink, however, earliest arrival transshipments do always exist [32]. This follows, for example, from the existence of lexicographically maximal flows in time-expanded networks; see, e.g., [28]. We refer to this problem as the *earliest arrival transshipment problem*. Hajek and Ogier [14] give the first polynomial time algorithm for the earliest arrival transshipment problem with zero transit times. Their algorithm uses $O(|V|)$ maximum flow computations. Fleischer [5] describes an algorithm that solves the problem in the same asymptotic time as one maximum flow computation by essentially reducing it to a parametric maximum flow problem that fits into the framework provided by Gallo, Grigoriadis, and Tarjan [12]. 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 [36] 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 and transshipments are motivated by applications related to evacuation. In the context of emergency evacuation from buildings, Berlin [2] and Chalmet et al. [4] study the quickest transshipment problem in networks with multiple sources and a single sink. Jarvis and Ratliff [22]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 [15] study an evacuation problem and propose solutions which further prevent 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 exactly and 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 a polynomial time 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 [24], we first recursively construct the earliest arrival pattern, that is, the piece-wise linear function that describes the time-dependent maximum flow value obeying supplies and demands. Similar to the algorithm of Hoppe and Tardos [17] for the quickest transshipment problem, our algorithm employs submodular function minimization within the parametric search framework of Megiddo [26, 27]. As a by-product, we present a new proof for the existence of earliest arrival transshipments that does not rely on time expansion. We finally show how to turn the earliest arrival pattern into an earliest arrival transshipment based on the quickest transshipment algorithm of Hoppe and Tardos [17].

Our algorithm can be interpreted as a non-trivial extension of Fleischer’s algorithm [5] for the special case of zero transit times. Similar to her approach, we identify a nested sequence of source nodes $S_0 \supseteq S_1 \supseteq \dots \supseteq S_k$ together with corresponding time bounds $\theta_0 < \theta_1 < \dots < \theta_k$ such that the sources in $S_i \setminus S_{i+1}$ have run empty at time θ_{i+1} . In order to find the subsets S_i and times θ_i for the case of general transit times, we need to employ parametric submodular function minimization while Fleischer uses a parametric max-flow computation. On inputs where all transit times are zero, the problems that our algorithm solves via submodular function minimization are indeed classical minimum cut problems.

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 shown in Section 4. We conclude with two open problems for future research in Section 5.

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 $v : S^+ \cup S^- \rightarrow \mathbb{R}$. For $\theta \geq 0$ and $S' \subseteq S^+ \cup S^-$ let $o^\theta(S')$ be the maximum amount of flow that can be sent from the sources $S^+ \cap S'$ to the sinks $S^- \setminus S'$ within time θ (ignoring supplies and demands).

We make use of the following result of Klinz [24]. It determines the feasibility of a given instance. We use here the continuous version which was extended from the discrete version of Klinz by Fleischer and Tardos [8].

³Strictly speaking, Jarvis and Ratliff [22] only consider the single-source case but their observation also applies to the more general case with multiple sources.

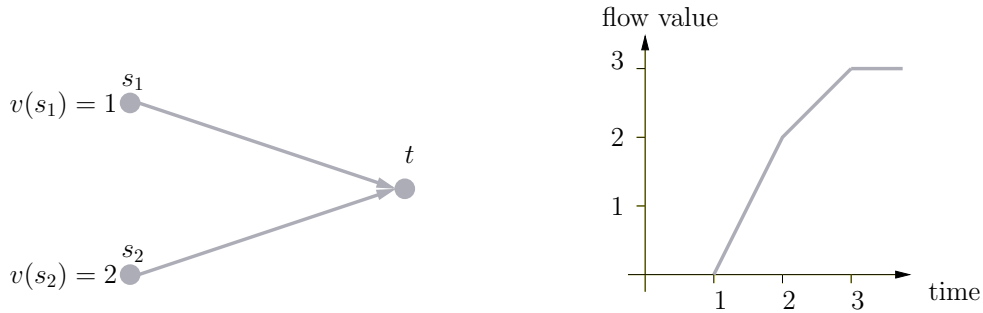


Figure 2: A simple example of a graph with two sources, unit capacities, and unit transit times where the optimal arrival pattern of a feasible earliest arrival transshipment is piecewise linear and non-convex.

LEMMA 2.1 (KLINZ [24]; FLEISCHER AND TARDOS [8]) *There exists a flow over time with time horizon θ that satisfies all supplies and demands if and only if*

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

For $\theta \geq 0$ and $S' \subseteq S^+ \cup S^-$, the value $o^\theta(S')$ can be obtained by a static min-cost flow computation. Consider the extended network \mathcal{N}' defined as follows. Starting from \mathcal{N} , introduce a super source s that is connected to all sources $S^+ \cap S'$ by an uncapacitated arc with transit time zero and a super sink t that can be reached from all sinks $S^- \setminus S'$ by such an arc. By construction of \mathcal{N}' , the value $o^\theta(S')$ is equal to the value of a maximal s - t -flow over time in \mathcal{N}' with time horizon θ . Further extend \mathcal{N}' by adding an uncapacitated dummy arc from t to s . It follows from the work of Ford and Fulkerson [9] that

$$o^\theta(S') = - \min \{ \text{cost}^\theta(x) \mid x \text{ circulation in } \mathcal{N}' \} . \quad (1)$$

Here, $\text{cost}^\theta(x)$ 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 $\tau_{(t,s)} := -\theta$. That is,

$$\text{cost}^\theta(x) = \sum_e \tau_e x_e$$

where the sum ranges over all arcs e in the extended network \mathcal{N}' . As a consequence of (1), the work of Ford and Fulkerson [9] implies that the function $\theta \mapsto o^\theta(S')$ is the negative of the cost function of a parametric min-cost flow problem. As such, it is well known to be piecewise linear and convex. (This follows, e.g., from a more general result on parametric linear optimization; see, e.g., [31, Sect. 6.5].)

Based on the work of Megiddo [25], Hoppe and Tardos [17] observe that the function $o^\theta : S^+ \cup S^- \rightarrow \mathbb{R}$ is submodular, that is,

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

for all $S', S'' \subseteq S^+ \cup S^-$.

For the rest of the paper we restrict to networks with a single sink t . The *earliest arrival pattern* $p : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ 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. Using this definition, 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 2.1 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 at node t 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. Recall that $o^\theta(S')$, $S' \subseteq S^+$, denotes the maximum amount of flow that can be sent out of sources in S' into sink t since $t \notin S'$.

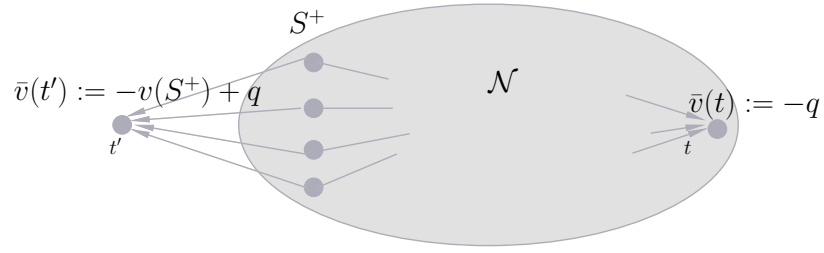


Figure 3: The extended network \mathcal{N}' used in the proof of Lemma 2.2. We use $v(t) := -q$ as an upper bound on the demanded flow that can enter the sink t . The remaining flow, namely $v(S^+) - q$ flow units, enters the additional sink t' that is connected from all sources via uncapacitated, zero transit time arcs.

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

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

PROOF. Consider an extended network \mathcal{N}' with an additional sink t' that can be reached from any source by an uncapacitated, zero transit time arc⁴. The demand of the new sink t' is defined to be $\bar{v}(t') := q - v(S^+)$ and the demand of the original sink t is set to $\bar{v}(t) := -q$. The supplies of sources remain unchanged, i.e., $\bar{v}(s) := v(s)$ for all $s \in S^+$. For $B \subseteq S^+ \cup \{t, t'\}$ let $\bar{o}^\theta(B)$ denote the maximum amount of flow that can be sent from the sources $S^+ \cap B$ to the sinks $\{t, t'\} \setminus B$. Notice that $p(\theta) \geq q$ if and only if the transshipment problem in the extended network \mathcal{N}' with time horizon θ and supplies and demands \bar{v} is feasible; see Figure 3. By Lemma 2.1 this is the case if and only if

$$\bar{o}^\theta(B) \geq \bar{v}(B) \quad \text{for all } B \subseteq S^+ \cup \{t, t'\}. \quad (3)$$

It remains to show that (2) holds if and only if (3) holds.

(2) \Leftarrow (3): By definition of \mathcal{N}' we get for $S' \subseteq S^+$

$$o^\theta(S') = \bar{o}^\theta(S' \cup \{t'\}) \stackrel{(3)}{\geq} \bar{v}(S' \cup \{t'\}) = v(S') + q - v(S^+) = q - v(S^+ \setminus S').$$

(2) \Rightarrow (3): Let $B \subseteq S^+ \cup \{t, t'\}$. We distinguish several cases. If $B \cap S^+ = \emptyset$, then $\bar{v}(B) \leq 0 \leq \bar{o}^\theta(B)$. We therefore assume from now on that $B \cap S^+ \neq \emptyset$. Since t' can be reached from every source node in S^+ by an uncapacitated arc with transit time zero, we get $o^\theta(B) = \infty > \bar{v}(B)$ if $t' \notin B$. We therefore assume from now on that $t' \in B$. If also $t \in B$, then $\bar{v}(B) \leq 0 \leq \bar{o}^\theta(B)$. It remains to consider the case that $B = S' \cup \{t'\}$ for some $S' \subseteq S^+$. In this setting we get

$$\bar{o}^\theta(B) = o^\theta(S') \stackrel{(2)}{\geq} q - v(S^+ \setminus S') = v(S') + q - v(S^+) = \bar{v}(B).$$

This concludes the proof. \square

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

COROLLARY 2.1 *The earliest arrival pattern p is piecewise linear.*

PROOF. As a result of Lemma 2.2 we get

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

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 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 depicted in the upper left corner of Figure 4 with unit transit times and unit capacities. The supplies of the sources are given in the figure.*

⁴A similar construction is described by Ford and Fulkerson [10] for the Caterer Problem.

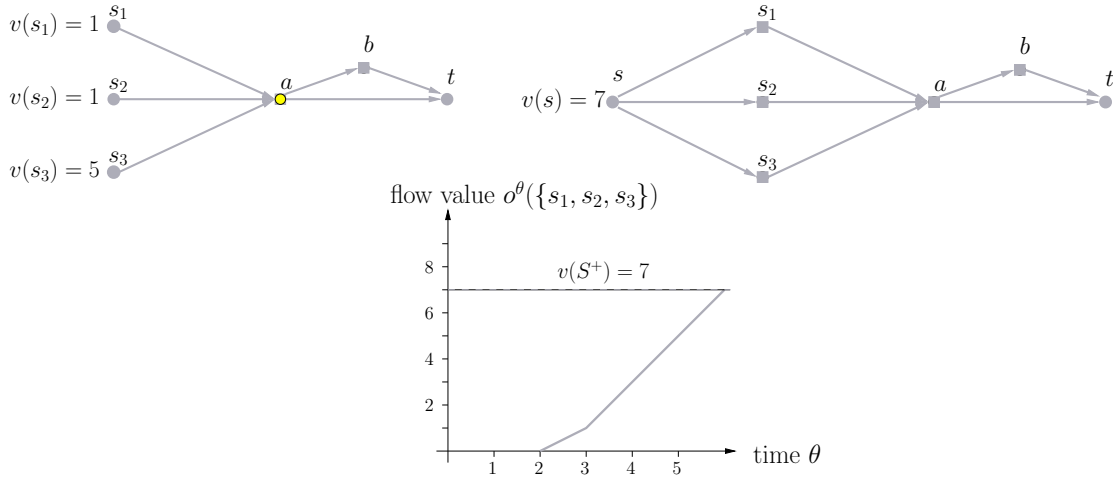


Figure 4: Example of a network $\mathcal{N} = (V, A)$, the network expanded by a supersource, and the corresponding s - t -earliest arrival pattern where individual supplies are ignored.

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 \mathcal{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 \mathcal{N}_0 and their entire supply $v(S^+)$ is shifted to the supersource s ; see Figure 4 for an example. Thus, a feasible s - t -flow over time in the extended network \mathcal{N}_0 induces a flow over time in \mathcal{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 \mathcal{N} might violate individual supplies at the source nodes.

The s - t -earliest arrival pattern in \mathcal{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 $\theta_1 := \sup\{\theta \mid p(\theta) = o^\theta(S^+)\}$.⁵ 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,*

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

PROOF. Consider an extended network \mathcal{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 \mathcal{N}' we get for $\bar{S} \subseteq S^+$ the following equalities:

$$\bar{o}^\theta(\bar{S}) = o^\theta(\bar{S}) \quad \text{and} \quad \bar{o}^\theta(\bar{S} \cup \{t\}) = o^{\theta'}(\bar{S}) . \quad (4)$$

We can now prove the statement of the lemma. By (4) and submodularity of $\bar{o}^\theta(\cdot)$ we get

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

This concludes the proof. \square

COROLLARY 3.1 *It holds that $p(\theta) = o^\theta(S^+)$ for all $0 \leq \theta \leq \theta_1$.*

⁵The supremum here is indeed a maximum since $p(\theta)$ and $o^\theta(S^+)$ are both continuous functions of θ and an upper bound for the time horizon is easily computable.

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

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

It follows from Lemma 3.1 that

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

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

Example. In order to compute the s - t -earliest arrival pattern for the network given in the left part of Figure 4 we insert a supersource s as depicted in the upper right corner of Figure 4. 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 lower part of Figure 4.

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 \mathcal{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 flow values over time determined by the 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 3.2 *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 . \square

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 with source set S_1 is denoted by p' . The following theorem is the main result of this section.

THEOREM 3.1 *Let $S_1 \subsetneq S^+$ be chosen according to Lemma 3.2 and p' the earliest arrival pattern of the modified instance with source set S_1 . Then,*

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

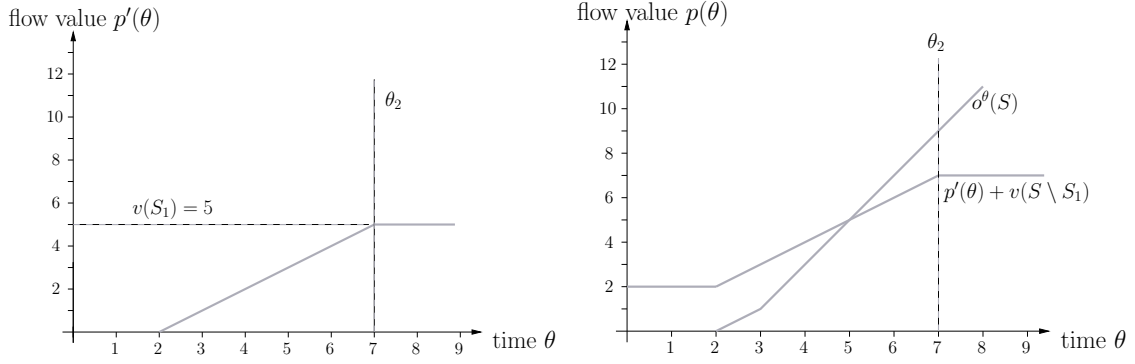


Figure 5: Optimal pattern p' for the problem with the reduced set of sources $S_1 = \{s_3\}$ (left) and the combined pattern p as the lower bound of the line segments (right).

PROOF. It follows from Corollary 3.1 that $p(\theta) = o^\theta(S^+)$ for $\theta \leq \theta_1$. It remains to show that

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

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:

$$\begin{aligned} o^\theta(S') &\geq o^\theta(S' \cap S_1) + o^\theta(S' \cup S_1) - o^\theta(S_1) && \text{by submodularity of } o^\theta(\cdot) \\ &\geq o^\theta(S' \cap S_1) + o^{\theta_1}(S' \cup S_1) - o^{\theta_1}(S_1) && \text{by Lemma 3.1} \\ &\geq \left(p'(\theta) - v(S_1 \setminus S')\right) + \left(o^{\theta_1}(S^+) - v(S^+ \setminus (S' \cup S_1))\right) && \text{by Lemma 2.2 and Lemma 3.2} \\ &\quad - \left(o^{\theta_1}(S^+) - v(S^+ \setminus S_1)\right) \\ &= p'(\theta) - v(S_1 \setminus S') - v(S^+ \setminus (S' \cup S_1)) + v(S^+ \setminus S_1) \\ &= p'(\theta) - v(S^+ \setminus S') + v(S^+ \setminus S_1). \end{aligned}$$

The result now follows from Lemma 2.2. \square

As a result of Theorem 3.1, 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 .

Example. Concerning our example given in Figure 4 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 lower part of Figure 4).

For the restricted earliest arrival transshipment problem with sources $S_1 = \{s_3\}$, the earliest arrival pattern p' is given in the left part of Figure 5. By Theorem 3.1, the resulting earliest arrival pattern p of the original instance is the lower envelope of the two functions depicted in the right part of Figure 5.

3.2 Computing the Earliest Arrival Pattern Theorem 3.1 reduces the problem of computing the earliest arrival pattern to an earliest arrival s - t -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 .

Algorithm 1: Computing the earliest arrival pattern.

Input: (G, S^+, t) **Output:** Earliest arrival pattern p .set $i := 0$, $S_i := S^+$, and $\theta_i := 0$;**1 while** $S_i \neq \emptyset$ **do****2** compute the maximal value $\theta_{i+1} \geq 0$ such that

$$o^{\theta_{i+1}}(S') \geq o^{\theta_{i+1}}(S_i) - v(S_i \setminus S') \quad \text{for all } S' \subseteq S_i;$$

3 compute an inclusion-wise minimal $S_{i+1} \subsetneq S_i$ with

$$o^{\theta_{i+1}}(S_{i+1}) = o^{\theta_{i+1}}(S_i) - v(S_i \setminus S_{i+1}); \quad (5)$$

4 compute the function $\theta \mapsto o^\theta(S_i)$ on the interval $[\theta_i, \theta_{i+1})$ and set

$$p(\theta) := o^\theta(S_i) + v(S^+ \setminus S_i) \quad \text{for } \theta \in [\theta_i, \theta_{i+1});$$

 $i := i + 1$;set $p(\theta) := v(S^+)$ for all $\theta \geq \theta_i$;

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 $S^+ \setminus S_1$ 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

$$\begin{aligned} o^{\theta_i}(S_{i+1}) &\leq o^{\theta_i}(S_i) + o^{\theta_{i+1}}(S_{i+1}) - o^{\theta_{i+1}}(S_i) && \text{by Lemma 3.1} \\ &= o^{\theta_i}(S_i) - v(S_i \setminus S_{i+1}) && \text{by (5)} \\ &= o^{\theta_i}(S_{i-1}) - v(S_{i-1} \setminus S_i) - v(S_i \setminus S_{i+1}) && \text{by (5) with } i := i - 1 \\ &= o^{\theta_i}(S_{i-1}) - v(S_{i-1} \setminus S_{i+1}) \end{aligned}$$

which contradicts the minimal choice of $S_i \supsetneq S_{i+1}$ in step 3 of the algorithm.

THEOREM 3.2 *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 4.

LEMMA 3.3 *For $0 \leq \theta_i \leq \theta_{i+1}$ and $S' \subseteq S^+$, the piecewise linear function $g : [\theta_i, \theta_{i+1}) \rightarrow \mathbb{R}$ 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 \mathcal{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 negative of the cost of a static min-cost circulation in \mathcal{N}' 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 static min-cost circulation x in \mathcal{N}' for $\theta = \theta_i$. Let \mathcal{N}'_x denote the residual network of x and let θ' be the length of a shortest s - t -path in \mathcal{N}'_x . Since there is the uncapacitated dummy arc (t, s) of cost $-\theta_i$ in \mathcal{N}'_x , optimality of x implies $\theta' \geq \theta_i$. Moreover, for all $\theta \in [\theta_i, \theta']$, the circulation x is still a static 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 \mathcal{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 \mathcal{N}''_x of the residual network \mathcal{N}'_x that is formed by all arcs that lie on some shortest s - t -path. Compute a maximum

static s - t -flow in \mathcal{N}_x'' and turn it into a circulation y in \mathcal{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 \mathcal{N}_x' , x the actual circulation, 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 static min-cost s - t -flow computation plus number of breakpoints many maximum static s - t -flow computations. \square

Example. In our example depicted in Figure 4 we can find the function $g : [\theta_i, \theta_{i+1}] \rightarrow \mathbb{R}$ as described above. For the interval $[\theta_0, \theta_1]$ we get the networks \mathcal{N}' , \mathcal{N}_x' , and \mathcal{N}_x'' as follows. (Remark that by convention given in Algorithm 1 we always set $\theta_0 = 0$.) Network \mathcal{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 $\tau_{(t,s)} = 0$. A static min-cost (maximum flow) circulation in this network is obviously the zero circulation. The residual network $\mathcal{N}_x'^{(1)}$ of \mathcal{N}' thus equals the original network \mathcal{N}' . A shortest path in $\mathcal{N}_x'^{(1)}$ for example is the path s, s_1, a, t and has length $\theta'^{(1)} = 2$. The network with zero flow and the shortest path found in the residual network is depicted in Figure 6.1. Next we need to consider subnetwork $\mathcal{N}_x''^{(1)}$ consisting of all arcs being part of some shortest s - t -path (see Figure 6.2). In this network we compute a maximum static s - t -flow. Such a flow is given by sending one unit of flow for example over path s, s_2, a, t . Reconsidering network $\mathcal{N}_x''^{(1)}$ together with a new circulation of one unit of flow along cycle s, s_2, a, t, s results in the new residual network $\mathcal{N}_x''^{(2)}$ which is depicted in Figure 6.3. The actual flow x is optimal for $\tau_{(t,s)} = -2$. There the shortest s - t -path, for example path s, s_3, a, b, t , has length $\theta'^{(2)} = 3$. In the subnetwork $\mathcal{N}_x''^{(2)}$ consisting of all arcs being part of some s - t -path of length 3 we compute again a maximum static s - t -flow. Such a path flow s, s_1, a, b, t is depicted in Figure 6.4. Sending one unit of flow along cycle s, s_1, a, b, t, s in network $\mathcal{N}_x''^{(2)}$ yields in network $\mathcal{N}_x''^{(3)}$ which is depicted in Figure 6.5. There no (shortest) s - t -path remains and therefore the value of $\theta'^{(3)}$ equals 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 6.6.

PROOF OF THEOREM 3.2. The correctness of the algorithm follows from Section 3.1 and in particular from Theorem 3.1. It thus remains to prove the stated bound on the running time of Algorithm 1.

First notice that the number of iterations of the while-loop in step 1 is bounded by the number of sources since at least one source is eliminated from S_i in every iteration. Since step 4 can be done in strongly polynomial time, it remains to show that steps 2 and 3 can also be done in strongly polynomial time.

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

$$f^\theta(S') \geq 0 \quad \text{for all } S' \subseteq S_i. \quad (6)$$

Since o^θ is submodular and the function

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

is modular, f^θ is submodular. According to equation (1), computing $f^\theta(S')$ for some $S' \subseteq S_i$ requires two static min-cost flow computations where the cost coefficients depend linearly on the parameter θ . It was shown by Grötschel, Lovász, and Schrijver [13] that there is a strongly polynomial algorithm for minimizing a submodular function. Combinatorial algorithms achieving strongly polynomial running time are given by Iwata, Fleischer, and Fujishige [21] and by Schrijver [33]. A fully combinatorial algorithm is given by Iwata [19]. It can therefore be tested in strongly polynomial time whether (6) is fulfilled for a fixed value θ . Thus, the maximum value of θ fulfilling (6) can be found in (weakly) polynomial time by embedding the submodular function minimization algorithm into a binary search framework. We can even achieve a running time bound for step 2 that is strongly polynomial in the input size if we replace binary search with Megiddo's parametric search framework (see [26, 27]). More details are given in Remark 3.1 below.

We finally discuss how to compute S_{i+1} in step 3 in strongly polynomial time. Notice that (5) 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 $f^{\theta_{i+1}}$,

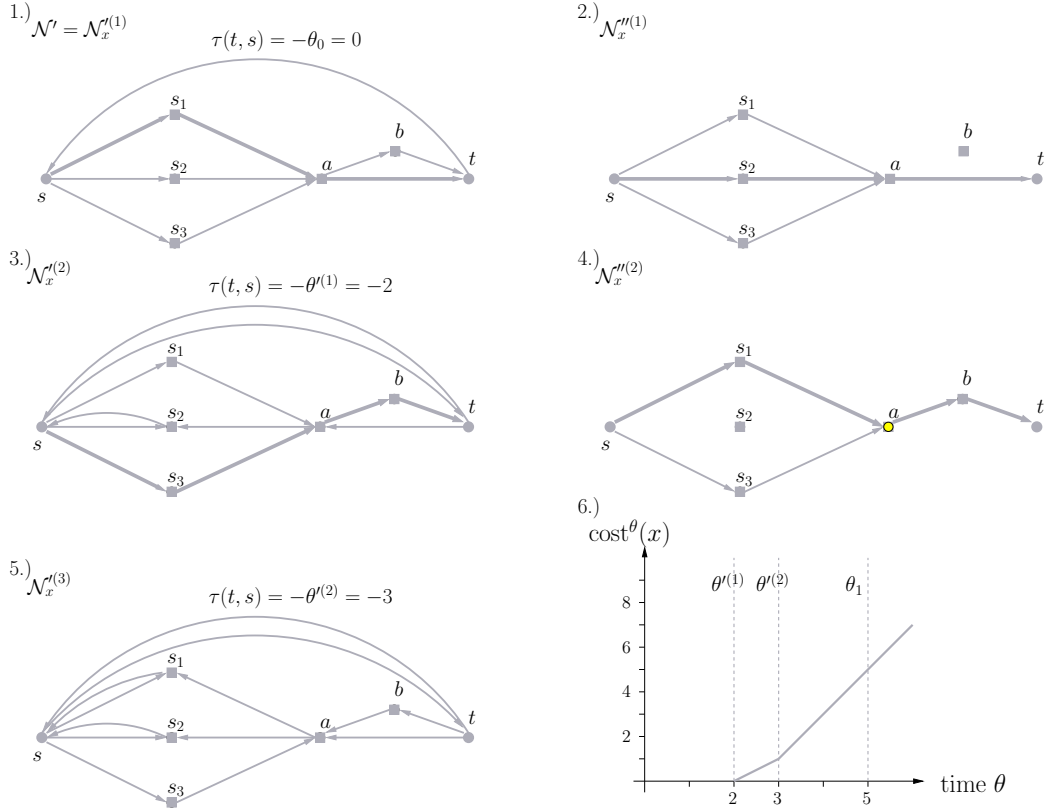


Figure 6: Networks used to compute the function $g : [\theta_0, \theta_1] \rightarrow \mathbb{R}$ as described in the proof of Lemma 3.3 for the network given in Figure 4.

there exists a unique inclusion-wise minimal subset S_{i+1} which can be obtained as follows⁶ (see, e.g., [34, 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 $f^{\theta_{i+1}}(S_{i+1}) = 0$. Nagano [29] even presents an efficient algorithm for submodular function minimization which provides a compact representation of all the minimizers. We can thus compute the desired subset S_{i+1} more efficiently with only one call to this algorithm. \square

In the following remark we shortly discuss the parametric search framework referred to in the proof above. In a second remark we point out that the running time of our algorithm can be further improved by employing a more specialized algorithm for parametric submodular function minimization.

REMARK 3.1 We give a rough outline of Megiddo's parametric search framework: Consider a fully combinatorial algorithm A for the non-parametric problem, that is, for standard submodular function minimization. A fully combinatorial algorithm uses only additions, subtractions, comparisons, and oracle calls for function values. For submodular function minimization, such an algorithm A is given by Iwata [19]. Algorithm A is now modified in order to solve the parametric problem. For this, the modified algorithm has to work with linear functions of the parameter θ instead of just constant numbers. Notice that adding or subtracting two linear functions yields a linear function again. Comparing two linear functions, however, imposes a problem. Whenever algorithm A compares two numbers, the modified version first has to determine whether the desired value θ_{i+1} is smaller or larger than the unique point θ at which the two linear functions cross (if at all). This can be decided by calling any algorithm B for submodular function minimization as a subroutine for the fixed parameter value θ .⁷ The number of calls of B is bounded by the number of comparisons performed by A which is strongly polynomial in the input size. In this way,

⁶For 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.

⁷For a more detailed but still intuitive description of Megiddo's parametric search in the context of flows over time we refer to [17, Sect. 5].

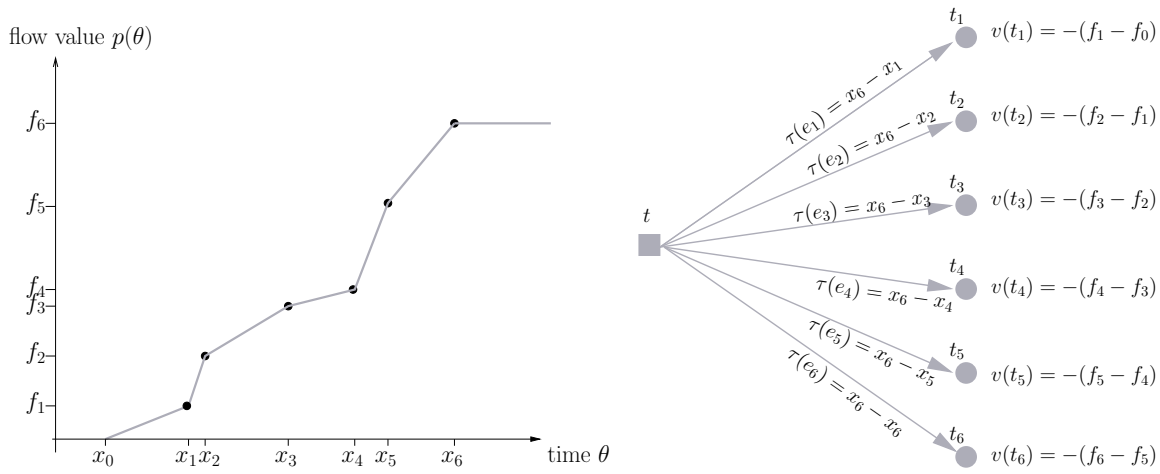


Figure 7: On the left hand side, we draw the earliest arrival pattern p with breakpoints (x_i, f_i) , $i = 1, 2, \dots, k = 6$. On the right hand side, the modified network is depicted. The capacity of arc $e_i = (t, t_i)$ is set to $(f_i - f_{i-1}) / (x_i - x_{i-1})$.

finding the desired value θ_{i+1} is reduced to a series of submodular function minimizations that decide (6) for fixed values of θ .

REMARK 3.2 As explained in Remark 3.1, the problem of computing θ_{i+1} boils down to a sequence of submodular function minimizations for certain values $\theta^1, \theta^2, \dots, \theta^k$ of the parameter θ . Instead of calling submodular function minimization algorithm B with input f^{θ^i} and thus computing the minimum of f^{θ^i} from scratch, for $i = 1, \dots, k$, one can get an improved running time as follows. It is not difficult to observe that our parametric submodular functions $f^\theta(\cdot)$ fulfill the *increasing difference condition*⁸, that is,

$$f^\theta(S') - f^\theta(S'') \leq f^{\theta'}(S') - f^{\theta'}(S'') \quad \text{for all } S' \supseteq S'' \text{ and } \theta \leq \theta'.$$

Fleischer and Iwata [20] present an algorithm that minimizes a sequence of submodular functions fulfilling this condition within the same asymptotic running time as a single execution of their submodular function minimization algorithm. Nagano [29] applies the same idea to a faster algorithm for submodular function minimization presented by Orlin [30]. We omit any further details that are beyond the scope of this paper.

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 $(x_0, f_0), (x_1, f_1), \dots, (x_k, f_k)$, that is,

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

An illustration is given in Figure 7. 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

$$x_0 < x_1 < \dots < x_k$$

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,

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

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 \mathcal{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 7.

⁸In the literature, a sequence of submodular functions fulfilling this condition is also referred to as a *strong map sequence*.

Node t is no longer a sink but just an intermediate node of the modified network \mathcal{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 transshipment in \mathcal{N} with earliest arrival pattern p naturally induces a feasible transshipment over time with time horizon x_k satisfying all supplies and demands in \mathcal{N}'*

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

The reverse direction of Lemma 4.1 also holds.

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

PROOF. We must prove that the flow passing through node t in an arbitrary feasible transshipment over time in \mathcal{N}' with time horizon x_k meets the earliest arrival pattern p . Since, for $i = 1, \dots, k$, the capacity of arc e_i equals the derivative of the earliest arrival pattern p in the time interval $[x_{i-1}, x_i]$ and since the negative of the demand of sink t_i is equal to this capacity times $x_i - x_{i-1}$, it suffices to prove the following claim:

CLAIM 1 *For each $i = 1, \dots, k$ the following holds: In a feasible transshipment over time the demand of sink t_i is satisfied by flow being sent into arc e_i within time interval $[x_{i-1}, x_i]$.*

We prove this claim by induction on i . The case $i = 1$ is clear since no flow can arrive at node t before time x_0 and, due to the transit time $x_k - x_1$ of arc e_1 , flow being sent into arc e_1 after time x_1 arrives at t_1 after time x_k . In order to prove the claim for $i > 1$ notice that the total amount of flow passing through node t before time x_{i-1} is bounded from above by $p(x_{i-1}) = f_{i-1}$. Since this amount is equal to the negative of the total demand of sinks t_1, \dots, t_{i-1} , it follows by induction that all flow passing through t before time x_{i-1} must be used to satisfy the demands of t_1, \dots, t_{i-1} . Thus, no flow is sent into arc e_i before time x_{i-1} . On the other hand, since the transit time of arc e_i is $x_k - x_i$, no flow is sent into this arc after time x_i . This concludes the proof. \square

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

LEMMA 4.3 *There exists a transshipment over time with time horizon x_k satisfying all supplies and demands in \mathcal{N}' .*

PROOF. We denote the set of sources in \mathcal{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 by time zero, 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 $\bar{\sigma}^\theta(S') \geq v(S')$ for $\theta = x_k$. This concludes the proof. \square

As a consequence we can state the following theorem.

THEOREM 4.1 *Given the earliest arrival pattern p with k breakpoints for network \mathcal{N} , an earliest arrival transshipment in \mathcal{N} can be obtained by computing a transshipment over time in a modified network \mathcal{N}' with k additional nodes and arcs.*

In order to compute a transshipment over time in the modified network \mathcal{N}' we can use the algorithm of Hoppe and Tardos [17]. Since the running time of this algorithm is bounded by a polynomial in the encoding size of the input \mathcal{N}' and since the encoding size of \mathcal{N}' is of the same order as the encoding size of \mathcal{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 \mathcal{N} .

5. Conclusions We conclude with two interesting open problems for future research. The second one was pointed out by one of the referees.

- (i) The algorithm for computing the earliest arrival pattern described in Section 3 only works on the network given as an input. In particular, it does not employ any form of time-expansion. Strictly speaking, this property does unfortunately not hold for the algorithm presented in Section 4. In order to turn the earliest arrival pattern into an earliest arrival transshipment, we use an extended network with additional sink nodes. In the worst case, the number of these sinks is exponential in the input size (but polynomial in the output size). It is an open problem to come up with an algorithm that only works on the given network and small (polynomial) extensions of it.
- (ii) As mentioned in the introduction, earliest arrival flows do in general not exist in networks with more than one sink. It is an open problem to classify multi-sink networks where earliest arrival flows do exist and provide efficient algorithms for computing them.

Acknowledgements. The authors thank Fritz Eisenbrand, Lisa Fleischer, Satoru Iwata, Bettina Klinz, Ekkehard Köhler, and Tom McCormick for interesting and helpful discussions on the topic of this paper. They are also grateful to two anonymous referees whose many valuable remarks and comments helped to improve the presentation of the paper considerably.

This work was supported by DFG Research Center MATHEON in Berlin and by DFG Focus Program 1126, “Algorithmic Aspects of Large and Complex Networks”, grant no. SK 58/4-1 and SK 58/5-3.

A preliminary version of this work appeared in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science [1].

References

- [1] N. Baumann and M. Skutella, *Solving evacuation problems efficiently: Earliest arrival flows with multiple sources*, Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (Berkeley, CA), 2006, pp. 399–408.
- [2] G. N. Berlin, *The use of directed routes for assessing escape potential*, National Fire Protection Association, Boston, MA, 1979.
- [3] R. G. Busaker and P. J. Gowen, *A procedure for determining minimal-cost network flow patterns*, Tech. Report 15, Operational Research Office, John Hopkins University, Baltimore, MD, 1961.
- [4] L. G. Chalmet, R. L. Francis, and P. B. Saunders, *Network models for building evacuation*, Management Science **28** (1982), 86–105.
- [5] L. K. Fleischer, *Faster algorithms for the quickest transshipment problem*, SIAM Journal on Optimization **12** (2001), 18–35.

- [6] L. K. Fleischer and M. Skutella, *The quickest multicommodity flow problem*, Integer Programming and Combinatorial Optimization (W. J. Cook and A. S. Schulz, eds.), Lecture Notes in Computer Science, vol. 2337, Springer, Berlin, 2002, pp. 36–53.
- [7] _____, *Quickest flows over time*, SIAM Journal on Computing **36** (2007), 1600–1630.
- [8] L. K. Fleischer and É. Tardos, *Efficient continuous-time dynamic network flow algorithms*, Operations Research Letters **23** (1998), 71–80.
- [9] L. R. Ford and D. R. Fulkerson, *Constructing maximal dynamic flows from static flows*, Operations Research **6** (1958), 419–433.
- [10] _____, *Flows in networks*, Princeton University Press, Princeton, NJ, 1962.
- [11] D. Gale, *Transient flows in networks*, Michigan Mathematical Journal **6** (1959), 59–63.
- [12] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan, *A fast parametric maximum flow algorithm and applications*, SIAM Journal on Computing **18** (1989), 30–55.
- [13] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric Algorithms and Combinatorial Optimization*, Algorithms and Combinatorics, vol. 2, Springer, Berlin, 1988.
- [14] B. Hajek and R. G. Ogier, *Optimal dynamic routing in communication networks with continuous traffic*, Networks **14** (1984), 457–487.
- [15] H. W. Hamacher and S. Tifecki, *On the use of lexicographic min cost flows in evacuation modeling*, Naval Research Logistics **34** (1987), 487–503.
- [16] B. Hoppe and É. Tardos, *Polynomial time algorithms for some evacuation problems*, Proceedings of the 5th Annual ACM–SIAM Symposium on Discrete Algorithms (Arlington, VA), 1994, pp. 433–441.
- [17] _____, *The quickest transshipment problem*, Mathematics of Operations Research **25** (2000), 36–62.
- [18] M. Iri, *A new method of solving transportation-network problems*, Journal of the Operations Research Society of Japan **26** (1960), 27–87.
- [19] S. Iwata, *A fully combinatorial algorithm for submodular function minimization*, Journal of Combinatorial Theory, Ser. B **84** (2002), 203–212.
- [20] S. Iwata and L. Fleischer, *A push-relabel framework for submodular function minimization and applications to parametric optimization*, Discrete Applied Mathematics **131** (2003), 311–322.
- [21] S. Iwata, L. Fleischer, and S. Fujishige, *A combinatorial strongly polynomial algorithm for minimizing submodular functions*, Journal of the ACM **48** (2001), 761–777.
- [22] J.J. Jarvis and H.D. Ratliff, *Some equivalent objectives for dynamic network flow problems*, Management Science **28** (1982), 106–108.
- [23] P. A. Jewel, *Optimal flow through networks*, Tech. Report 8, Operations Research Center, MIT, Cambridge, MA, 1958.
- [24] B. Klinz, Cited as personal communication (1994) in [17].
- [25] N. Megiddo, *Optimal flows in networks with multiple sources and sinks*, Mathematical Programming **7** (1974), 97–107.
- [26] _____, *Combinatorial optimization with rational objective functions*, Mathematics of Operations Research **4** (1979), 414–424.
- [27] _____, *Applying parallel computation algorithms in the design of serial algorithms*, Journal of the ACM **30** (1983), 852–865.
- [28] E. Minieka, *Maximal, lexicographic, and dynamic network flows*, Operations Research **21** (1973), 517–527.
- [29] K. Nagano, *A faster parametric submodular function minimization algorithm and applications*, Mathematical Engineering Technical Reports METR 2007-43, University of Tokyo, 2007.
- [30] J. B. Orlin, *A faster strongly polynomial time algorithm for submodular function minimization*, Integer Programming and Combinatorial Optimization (M. Fischetti and D. P. Williamson, eds.), Lecture Notes in Computer Science, vol. 4513, Springer, Berlin, 2007, pp. 240–251.
- [31] M. Padberg, *Linear optimization and extensions*, Springer, Berlin, 1995.
- [32] D. Richardson and É. Tardos, Cited as personal communication (2002) in [7].
- [33] A. Schrijver, *A combinatorial algorithm minimizing submodular functions in strongly polynomial time*, Journal of Combinatorial Theory, Series B **80** (2000), 346–355.

- [34] _____, *Combinatorial optimization: Polyhedra and efficiency*, Springer, Berlin, 2003.
- [35] M. Skutella, *An introduction to network flows over time*, Research Trends in Combinatorial Optimization (W. Cook, L. Lovász, and J. Vygen, eds.), Springer, Berlin, 2009, pp. 451–482.
- [36] S. Tjandra, *Dynamic network optimization with application to the evacuation problem*, Ph.D. thesis, Universität Kaiserslautern, Shaker Verlag, Aachen, 2003.
- [37] W. L. Wilkinson, *An algorithm for universal maximal dynamic flows in a network*, Operations Research **19** (1971), 1602–1612.