Minimum-Cost Dynamic Flows: The Series-Parallel Case

Bettina Klinz
TU Graz, Institut für Mathematik B, Steyrergasse 30, A-8010 Graz, Austria

Gerhard J. Woeginger
Department of Mathematics and Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands

A dynamic network consists of a directed graph with capacities, costs, and integral transit times on the arcs. In the *minimum-cost dynamic flow problem* (MCDFP), the goal is to compute, for a given dynamic network with source *s*, sink *t*, and two integers *v* and *T*, a feasible dynamic flow from *s* to *t* of value *v*, obeying the time bound *T*, and having minimum total cost. MCDFP contains as subproblems the *minimum-cost maximum dynamic flow problem,* where *v* is fixed to the maximum amount of flow that can be sent from *s* to *t* within time *T* and the *minimum-cost quickest flow problem,* where is *T* is fixed to the minimum time needed for sending *v* units of flow from *s* to *t*. We first prove that both subproblems are NP-hard even on two-terminal series-parallel graphs with unit capacities. As main result, we formulate a greedy algorithm for MCDFP and provide a full characterization via forbidden subgraphs of the class $\mathcal{G}$ of graphs, for which this greedy algorithm always yields an optimum solution (for arbitrary choices of problem parameters). $\mathcal{G}$ is a subclass of the class of two-terminal series-parallel graphs. We show that the greedy algorithm solves MCDFP restricted to graphs in $\mathcal{G}$ in polynomial time. © 2004 Wiley Periodicals, Inc.

Keywords: minimum-cost flow; dynamic network flow; greedy algorithm; two-terminal series parallel networks

1. INTRODUCTION

A dynamic network is defined by a directed graph $G = (N, A)$ with sources, sinks, and nonnegative capacities u_a, costs c_a, and nonnegative integral transit times τ_a for every arc $a \in A$. In a feasible dynamic flow, at most u_a units of flow can be sent along the arc a within each integral time step. The flow which leaves the tail of arc a at time θ and is sent along arc a reaches the head of a at time $\theta + \tau_a$.

The minimum-cost dynamic flow problem is defined by a dynamic network with a single source s, a single sink t, and two nonnegative integers, the time bound T and the flow value v. All flow units sent from the source s must arrive at the sink t no later than at time T. The minimum-cost dynamic flow problem is to find a feasible dynamic flow f sending v units of flow from s to t in such a way that the cost of f over all time steps is minimum.

Two special cases of the minimum-cost dynamic flow problem are obtained if T and v are not chosen independently. In the minimum-cost maximum dynamic flow problem, v is set to the maximum value of a feasible dynamic flow with time bound T, and in the minimum-cost quickest flow problem, T is set to the minimum time such that there exists a feasible dynamic flow with value v and time bound T.

Dynamic flow problems arise in many applications, for example, in production–distribution systems, communications systems, truck and railway scheduling, and building evacuation problems (see the surveys by Aronson [1] and Powell et al. [16] for further details). Recently, some authors (see, e.g., [5, 6]) preferred to use the term *flows over time* instead of the term *dynamic flows* to avoid misunderstandings with the notion of dynamic graph problems used by theoretical computer scientists.

1.1. Previous and Related Work

The minimum-cost dynamic flow problem is equivalent to a traditional minimum-cost flow problem on a related, exponentially large time-expanded graph $G(T)$. Most known methods for solving the minimum-cost dynamic flow model work directly on $G(T)$ and, thus, their running time is only pseudopolynomial (as it depends polynomially on T rather than on $\log T$).

An alternative solution method which takes into account the multiperiod structure of the problem was developed by

Received November 2001; accepted October 2003
Correspondence to: B. Klinz; e-mail: klinz@opt.math.tu-graz.ac.at
Contract grant sponsor: Christian Doppler Laboratorium für Diskrete Optimierung
Contract grant sponsor: Fonds zur Förderung der wissenschaftlichen Forschung; contract grant number: P8971-PHY
Contract grant sponsor: START Project Y43-MAT of the Austrian Ministry of Science
A preliminary version of this paper appeared in the Proceedings of the 4th International Conference on Integer Programming and Combinatorial Optimization, Springer Lecture Notes in Computer Science 920, 1995, pp. 329–343

Aronson and Chen [2]. Their forward-network simplex procedure finds a minimum-cost dynamic flow even in the more general case of time-dependent capacities and costs. However, this method again is not polynomial and thus fails as T becomes large.

If all arc costs are zero, the minimum-cost maximum dynamic flow problem turns into the well-known maximum dynamic flow problem and the minimum-cost quickest flow problem turns into the quickest-flow problem. Both problems can be solved in (strongly) polynomial time (see Ford and Fulkerson [7] and Burkard et al. [4]). Multiterminal generalizations of these problems were considered in Hoppe and Tardos [10, 11].

In the multicommodity case, already the versions without costs turn out to be NP-hard (see Hall et al. [9]). It is not even known whether there always exists an optimal solution which can be described in polynomial space.

A different version of the minimum-cost dynamic flow problem was considered in Orlin [14]. In his model, the time horizon is infinite, that is, $T = \infty$, and flow conservation is required only for all time steps $\theta \geq \max_{a \in A} \tau_a$. Each arc a is assigned a convex cost function and the aim is to find a feasible infinite-horizon flow minimizing the average cost per period. Orlin showed that such a flow can be found in polynomial time. This result relies on the property that there always exists a stationary optimal flow, that is, the flow on the arcs does not change as the time progresses. No equivalent result holds in the finite horizon case.

Since many dynamic flow problems are known to be NP-hard, the issue of approximability plays an important role. Recently, Fleischer and Skutella [5, 6] obtained fully polynomial time-approximation schemes for the minimum-cost quickest flow problem and the quickest multicommodity flow problem. The main idea behind these approximation algorithms is the concept of a condensed time-expanded network where a coarser discretization is used than for the full time-expanded network.

1.2. Organization of the Paper

In Section 2, we give a formal description of the minimum-cost dynamic flow problem and introduce the basic notation used throughout this paper. In Section 3, we prove that both the minimum-cost quickest flow problem and the minimum-cost maximum dynamic flow problem are NP-hard even for two-terminal series-parallel graphs with unit capacities. In Section 4, we propose a greedy algorithm and provide an exact characterization of the class of graphs for which this approach yields an optimal solution. The paper is concluded with a short discussion in Section 5.

2. DEFINITIONS AND PRELIMINARIES

2.1. Dynamic Networks and Dynamic Flows

Let $G = (N, A)$ be a directed (multi) graph with node set N and arc set A where each arc $a \in A$ is characterized by its *tail* $t(a)$ and its *head* $h(a)$; a is directed from $t(a)$ to $h(a)$. For each node i, we denote the set of arcs a with $t(a) = i$ by $A^+(i)$ and the set of arcs a with $h(a) = i$ by $A^-(i)$. If there is a single arc a with $t(a) = i$ and $h(a) = j$, this arc will also be referred to by (i, j).

A *path* P in G from node j to node k is an alternating sequence of nodes and arcs such that $P = (i_0, a_1, i_1, a_2 \cdots a_p, i_p)$, $i_0 = j$, and $i_p = k$, and for each $r = 1, \ldots, p$, either arc a_r has head i_r and tail i_{r-1} or else it has head i_{r-1} and tail i_r. In the former case, the arc is called a *forward arc* of the path; in the latter case, it is called a *backward arc.* A path is called *directed* if every arc is a forward arc and *simple* if no node is repeated.

A *cycle* is a path for which the initial node i_0 coincides with the final node i_p. G is said to be *acyclic* if it does not contain a *directed cycle,* that is, a cycle with forward arcs only.

A *dynamic network* $\mathcal{N} = (G, u, \tau, c, s, t)$ consists of a directed (multi) graph G with *source* s and *sink* t and three numbers attached to each arc $a \in A$, namely, a nonnegative real *capacity* u_a, a nonnegative integer *transit time* τ_a, and a *cost* c_a. Given a directed path $P = (i_0, a_1, i_1, a_2 \cdots a_p, i_p)$, let $\tau(P) = \sum_{r=1}^{p} \tau_{a_r}$ denote its transit time, and $c(P) = \sum_{r=1}^{p} c_{a_r}$, its cost. The cost of an arbitrary path and, respectively, cycle, is given by the sum of the cost of its forward arcs minus the sum of the cost of its backward arcs.

For simplicity, we assume that no arc enters the source s and no arc leaves the sink t. Moreover, we assume that there exists a directed path in G from the source s to any node i, $i \neq s$, as well as a directed path from any node i, $i \neq t$ to the sink t.

Let $f_a(\theta)$ denote the flow which leaves the tail node $t(a)$ at time θ along the arc a. This flow arrives at the head node $h(a)$ at time $\theta + \tau_a$. To allow the flow to arrive in an inner node $i \neq s, t$ at time θ_1, then wait there for some time and leave node i again at time $\theta_2 > \theta_1$, we introduce so-called *holdover arcs* modeled by loops a with $h(a) = t(a) = i$. These loops get a transit time of one, infinite capacity and an arbitrary nonnegative cost which can be chosen depending on what is needed in the actual application. (In many cases, zero-cost holdovers are most natural.)

For notational convenience, let I_T denote the set $\{0, 1, \ldots, T\}$ and let A_0 denote the union of the original arc set A and the set of loops introduced for modeling holdovers. Following Ford and Fulkerson [7], the mapping $f : A_0 \times \{0, \ldots, T\} \rightarrow \mathbb{R}_0^+$ is said to be a *feasible dynamic flow* (or just dynamic flow) if the following two groups of constraints are satisfied:

$$\sum_{\substack{a \in A_0 \\ t(a)=i}} f_a(\theta) - \sum_{\substack{a \in A_0 \\ h(a)=i}} f_a(\theta - \tau_a) = 0 \quad \text{for all } i \in N \backslash \{s, t\}, \theta \in I_T \tag{1}$$

$$0 \leq f_a(\theta) \leq u_a \quad \text{for all } a \in A_0, \theta \in I_T, \tag{2}$$

where, for notational convenience, we assume throughout that $f_a(\theta) = 0$ for $\theta < 0$. Equations (1) require that, for any time $\theta \in \{0, \ldots, T\}$ and any node $i \neq s, t$, the amount of flow which enters node i at time θ equals the amount of flow which leaves node i at time θ. The inequalities (2) require that the capacity constraints are fulfilled for each arc $a \in A$ and each time that $\theta \in \{0, \ldots, T\}$.

Note that the flow conservation constraints (1) imply that the net amount which leaves the source equals the net amount which enters the sink, that is,

$$\sum_{\theta=0}^{T} \sum_{a \in A^+(s)} f_a(\theta) = \sum_{\theta=0}^{T} \sum_{a \in A^-(t)} f_a(\theta - \tau_a) =: v^f. \tag{3}$$

v^f is called the *value* of the dynamic flow f.

2.2. Problem Statement

Given a dynamic network $\mathcal{N} = (G, u, \tau, c, s, t)$, a nonnegative integral time bound T, and a nonnegative integral flow value v, the *minimum-cost dynamic flow problem* (MCDFP) is to determine a feasible dynamic flow f with value $v^f = v$ and minimum total cost c^f, where

$$c^f = \sum_{\theta=0}^{T} \sum_{a \in A_0} c_a f_a(\theta). \tag{4}$$

The *minimum-cost maximum dynamic flow problem* results from MCDFP by setting v to the maximum value $v_{\max}(T)$ of a feasible dynamic flow with time bound T.

The *minimum-cost quickest flow problem* results if, for given v, the time-bound T is set to the minimum time $T_{\min}(v)$ such that there exists a feasible dynamic flow with time bound $T_{\min}(v)$ and value v.

2.3. Static Flows and Time-expanded Graphs

To distinguish traditional flows in G from dynamic flows, traditional flows will, henceforth, be referred to as *static flows.* A mapping $g : A \mapsto \mathbb{R}_0^+$ is said to be a (*feasible*) *static* s, t-*flow* in $G = (N, A)$ if g satisfies the flow conservation constraint $\sum_{a \in A^+(i)} g_a - \sum_{a \in A^-(i)} g_a = 0$ for every node $i \neq s, t$ and the capacity constraint $0 \leq g_a \leq u_a$ for every arc $a \in A$. A path P from s to t in G is said to be an *augmenting path* with respect to the flow g if $g_a < u_a$ holds for all forward arcs a of path P and $g_{a'} > 0$ holds for all backward arcs a' of P. A *minimum-cost augmenting path* is an augmenting path with minimum cost.

It is easy to see that a minimum-cost dynamic flow in G corresponds to a minimum-cost static flow in an enlarged graph, the so-called *time-expanded graph* $G(T) = (N(T), A(T))$, which is defined as follows: For each node $i \in N$, we introduce $T + 1$ copies in $N(T)$, denoted by $i(0), \ldots, i(T)$, and for each arc $a \in A_0$, we introduce $\max\{0, T - \tau_a + 1\}$ copies, with tail $i(\theta)$ and head $j(\theta + \tau_a)$, $\theta = 0, \ldots, \max\{0, T - \tau_a + 1\}$, where i is the tail of the original arc a and j is the head of a. Each copy of arc a is assigned a capacity of u_a and a cost of c_a. Note that if the loop (i, i) is contained in A_0 then $A(T)$ contains the arcs $(i(\theta), i(\theta + 1))$ for $\theta = 0, \ldots, T - 1$. These arcs are called *holdover arcs.*

In addition, we add to $N(T)$ a super source s' and a super sink t' and link s' to all copies of s and all copies of t to t'. These arcs all get zero cost and infinite capacity. A feasible dynamic flow from s to t in the graph G corresponds to a feasible static flow from s' to t' in the time-expanded graph $G(T)$.

2.4. Temporally Repeated Flows

The size of the time-expanded network increases with the size of the time bound T and thus is exponentially large in the size of the input. This raises the question of finding ways for computing and representing dynamic flows that are more efficient than the time-expanded network approach. To that end, Ford and Fulkerson [7] introduced a special class of dynamic flows with a particularly simple representation, namely, the class $\mathcal{T}$ of *temporally repeated flows* which are defined as follows:

Let g be a feasible static s, t-flow and decompose g into a sum of path flows $g^{(1)}, \ldots, g^{(q)}$ along directed simple s, t-paths $P_1, \ldots, P_q$ (cycles are omitted). Let $\Gamma := \{g^{(1)}, \ldots, g^{(q)}\}$. Given a time bound $T \geq \max_{r=1,\ldots,q} \tau(P_r)$, repeat each path flow $g^{(r)} \in \Gamma$ exactly $(T + 1 - \tau(P_r))$ times. The resulting dynamic flow $f^{g,\Gamma,T}$ is a feasible dynamic flow and is referred to as *temporally repeated flow* which is induced by the static flow g and its path flow decomposition Γ.

It is easy to check that the value of the temporally repeated flow $f^{g,\Gamma,T}$ is given by

$$\sum_{r=1}^{q} (T + 1 - \tau(P_r)) \cdot |g^{(r)}| = (T + 1)|g| - \sum_{a \in A} \tau_a g_a, \tag{5}$$

where $|g|$ denotes the value of the static flow g and $|g^{(r)}|$, $r = 1, \ldots, q$, denotes the value of the path flow $g^{(r)}$. Note that the expression on the right-hand side of (5) only depends on g and is independent of the set of path flows Γ.

In both the maximum dynamic flow problem and the quickest flow problem, the search for an optimal flow can be restricted to the class of temporally repeated flows (cf. Ford and Fulkerson [7] and Burkard et al. [4], respectively). Hence, one might hope that this remains true if arc costs are added to the problem. The following simple example demonstrates that this is not the case.

Example. *Consider the graph with node set* $N = \{s, 1, t\}$ *and arc set* $A = \{a_1, a_2, a_3\}$, *where* $a_1 = (s, 1)$, $t(a_2) = t(a_3) = 1$, *and* $h(a_2) = h(a_3) = t$. *Let* $\tau_{a_2} = c_{a_3} = 1$ *and let all other transit times and costs be zero and all capacities be one. Let* $T = 1$ *and* $v = 2$ [*note that* $T = T_{\min}(v)$ *and* $v = v_{max}(T)$, *respectively*]. *The only feasible temporally re-*

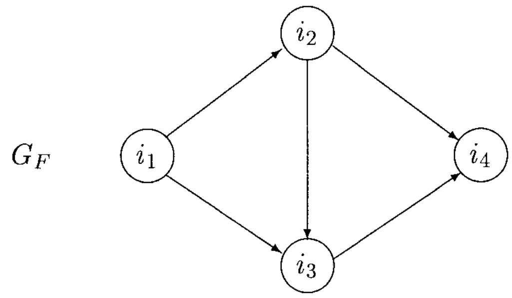

FIG. 1. The forbidden subgraph G_F for two-terminal series-parallel graphs.

peated flow f_1 sends one unit of flow along the path $P_1 = (s, a_1, 1, a_3, t)$ at $\theta = 0, 1$. Using path $P_2 = (s, a_1, 1, a_2, t)$ at $\theta = 0$ and path P_1 only at $\theta = 1$ results in a feasible dynamic flow f_2 with cost $c^{f_2} = 1$ while $c^{f_1} = 2$.

2.5. Two-terminal Series-parallel Graphs

A *two-terminal series-parallel graph* with source s and sink t is a directed (multi) graph $G = (N, A)$ which either consists of the single arc (s, t) or can obtained recursively as follows: If G_1 and G_2 are two-terminal series-parallel graphs with sources s_1 and, respectively, s_2 and sinks t_1 and, respectively, t_2, then the graph that is obtained by one of the following operations is also two-terminal series-parallel:

(a) Parallel composition $G_1 \parallel G_2$: Identify the source s_1 of G_1 with the source s_2 of G_2 and the sink of G_1 with the sink of G_2. The common source is the source of the composition and the common sink is its sink.
(b) Series composition $G_1 \circ G_2$: Identify the sink t_1 of G_1 with the source s_2 of G_2. The source of G_1 is the source of the composition. The sink of the composition is the sink of G_2.

A linear time recognition algorithm for two-terminal series-parallel graphs was given in Valdes et al. [17].

Two-terminal series-parallel graphs can also be characterized via forbidden homeomorphic subgraphs. A graph G contains a subgraph *homeomorphic* to a graph G', if G' can be obtained from G by a sequence of the following operations: (O1) remove an arc; (O2) remove an isolated node; (O3) if a node i has in-degree one and out-degree one, delete i and replace the two arcs (k, i) and (i, j) by the new arc (k, j).

Lemma 2.1 (see e.g., Valdes et al. [17]). *An acyclic directed graph G is two-terminal series-parallel if and only if it does not contain a subgraph homeomorphic to the graph G_F in Figure 1.*

3. THE COMPLEXITY OF THE MCDFP

It is easy to see that the general MCDFP is NP-hard: Finding a minimum-cost dynamic flow of value $v = 1$ amounts to finding a minimum-cost s, t-path with transit time $\leq T$, that is, to solving a constrained shortest path problem which is NP-hard (see Garey and Johnson [8]). In the following, we will derive the stronger result that finding a minimum-cost quickest flow and, respectively, a minimum cost maximum dynamic flow is NP-hard.

Theorem 3.1. *The minimum-cost quickest flow problem and the minimum-cost maximum dynamic flow problem, respectively, are NP-hard for two-terminal series-parallel graphs with unit capacities.*

Proof. The proof is done by a reduction from the NP-complete even–odd partition problem EOP (cf. Garey and Johnson [8]): Given $2d$ positive integers β_r, $r = 1, \ldots, 2d$, such that $\sum_{r=1}^{2d} \beta_r = 2B$, does there exist a set $I \subset \{1, \ldots, 2d\}$ such that $\sum_{r \in I} \beta_r = B$ and $|I \cap \{2h - 1, 2h\}| = 1$ for all $h = 1, \ldots, d$?

Let an instance of EOP be given. The corresponding dynamic flow problem is constructed as follows: First, we set up a two-terminal series-parallel graph $G = (N, A)$ with node set $N := \{1, 2, \ldots, d + 2\}$, source $s = 1$, sink $t = d + 2$, and arc set A, where for each $h = 1, \ldots, d$ the arc set A contains two parallel arcs a_{2h-1} and a_{2h}, both with tail h and head $h + 1$. In addition, we add the arcs $a_{2d+1} = (1, d + 1)$ and $a_{2d+2} = (d + 1, d + 2)$. For each $h = 1, \ldots, d$, the arc a_{2h-1} has transit time β_{2h-1} and cost β_{2h} and the arc a_{2h} has transit time β_{2h} and cost β_{2h-1}. Furthermore, arc a_{2d+1} has transit time 0 and cost $B + 1$ and arc a_{2d+2} has zero transit time and zero cost. Finally, we set all capacities to one and define $C = B^2 + 2B$, $v = B + 1$ and $T = B$ [note that $v = v_{\max}(T)$ and $T = T_{\min}(v)$]

Claim. *There exists a feasible dynamic flow of value v and cost $\leq C$ if and only if the above graph G contains a directed s, t-path P with transit time $\tau(P) \leq B$ and cost $c(P) \leq B$.*

Proof of the claim. Since $T = B$, paths P from s to t with $\tau(P) > B$ are of no use. If all v units of flow are sent along s, t-paths with cost $>B$, then the cost of the resulting dynamic flow will become at least $(B + 1)(B + 1) = B^2 + 2B + 1 > C$. Consequently, a feasible dynamic flow of value v and cost $\leq C$ can exist only if there exists an s, t-path P with $\tau(P) \leq B$ and $c(P) \leq B$.

Conversely, given an s, t-path P with $\tau(P) \leq B$ and $c(P) \leq B$, construct the following dynamic flow f: At time $T - \tau(P) = B - \tau(P)$, dispatch one unit of flow along the path P, and at time steps $\theta = 0, \ldots, B - 1$, dispatch one unit of flow along the path $P' = (s, a_{2d+1}, d + 1, a_{2d+2}, t)$. f is feasible, has value v, and total cost $c^f = c(P) + B(B + 1) \leq B^2 + 2B = C$.

Proof of Theorem 3.1, continued. It remains to be shown that there exists an s, t-path P with $\tau(P) \le B$ and $C(P) \le B$ if and only if the given EOP instance is a "yes"-instance. To that end, note that any s, t-path P with $c(P) \le B$ must contain exactly one of the arcs a_{2h-1} and a_{2h} for each $h = 1, \ldots, d$. Hence, $\tau(P) + c(P) = \sum_{r=1}^{2d} \beta_r = 2B$, which, in turn, implies that the only s,t-paths P with $\tau(P) \le B$ and cost $c(P) \le B$ are those with $\tau(P) = B$ and $c(P) = B$. Such paths correspond in a straightforward manner to a solution of the EOP: For each $r = 1, \ldots, 2d$, index r is in set I if and only if arc a_r is in path P. ∎

Remark 3.1. *Note that the structure of the optimal flow in instance I constructed in the proof above is very simple; the optimal flow is either a temporally repeated flow or it deviates from a temporally repeated flow only by sending flow along an additional path P at a single time unit. Nevertheless, it is NP-hard to find an optimal flow.*

Although the class of temporally repeated flows does, in general, not contain a minimum-cost maximum dynamic flow (respectively, a minimum-cost quickest flow), in view of Theorem 3.1, it still might be interesting to find a temporally repeated flow with minimum cost. Unfortunately, it turns out that this problem is NP-hard as well. To prove this, we consider the *minimum-cost maximum temporally repeated flow problem* (MCMTRFP) which can be stated as follows:

MCMTRFP

Input. A dynamic network $\mathcal{N} = (G, u, \tau, c, s, t)$, a nonnegative integral time bound T, and a cost bound C. Let $v = v_{\max}(T)$ denote the maximum amount of flow that can be sent from s to t within time T.
Question. Does there exist a feasible *integral* static flow g and a decomposition Γ of g into *integral* path flows $g^{(1)}, \ldots, g^{(q)}$ such that the induced temporally repeated flow $f = f^{g,\Gamma,T}$ has flow value v and overall cost $c^f \le C$?

Theorem 3.2. *The MCMTRFP is strongly NP-hard already for two-terminal series-parallel graphs with unit capacities.*

Proof. The proof is done by a reduction from the following variant of 3-PARTITION which is strongly NP-hard (see Garey and Johnson [8] and Papadimitriou et al. [15]):

3-PARTITION

Input. Three sets of d positive integers each, $\{\alpha_1, \ldots, \alpha_d\}$, $\{\beta_1, \ldots, \beta_d\}$, and $\{\gamma_1, \ldots, \gamma_d\}$, each greater than $B/4$, where $B = (1/d) \sum_{r=1}^{d} (\alpha_r + \beta_r + \gamma_r)$.
Question. Do there exist two permutations ϕ and ψ of $\{1, \ldots, d\}$ such that all sums $\alpha_r + \beta_{\phi(r)} + \gamma_{\psi(r)}$, $r = 1, \ldots, d$, are equal to B?

Given an instance I_P of 3-PARTITION, we now construct an instance I_M of MCMTRFP as follows:

Define $\tau_{\max} := \max_{r=1,\ldots,d}\{\alpha_r, \beta_r, \gamma_r\}$. We introduce a graph $G = (N, A)$ with node set $N := \{s, 1, 2, t\}$, source s, sink t, and the following $3d$ arcs: For each $r = 1, \ldots, d$, there are three arcs a_r, a_{d+r}, and a_{2d+r}. The arc a_r has tail s, head 1, transit time α_r, and cost $\tau_{\max} - \alpha_r$; the arc a_{d+r} has tail 1, head 2, transit time β_r, and cost $\tau_{\max} - \beta_r$; and, finally, the arc a_{2d+r} has tail 2, head t, transit time γ_r, and cost $\tau_{\max} - \gamma_r$. The capacities of all arcs are set to 1. Let $M := \max\{\alpha_{r_1} + \beta_{r_2} + \gamma_{r_3};\ 1 \le r_1, r_2, r_3 \le d\}$ and $T := M + B - 1$. Finally, set $C_1 := 3d(T + 1)\tau_{\max} - dB(3\tau_{\max} + T + 1)$ and $C := C_1 + dB^2$.

It remains to be shown that the instance I_M of MCMTRFP constructed above is a "yes"-instance if and only if the original instance I_P of 3-PARTITION is a "yes"-instance. To that end, the following observations turn out to be essential:

(i) First, observe that $v_{\max}(T) = dM$, that is, in instance I_M, at most dM units of flow can be sent from s to t within time $T = M + B - 1$. To see this, let $g_{\max}$ denote the static flow which sends one unit of flow along each arc $(i, j) \in A$. Obviously, $g_{\max}$ has value d and is the unique maximum static s,t-flow in G. Using formula (5), it follows that the value of any temporally repeated flow that is induced by $g_{\max}$ is equal to $(M + B)d - \sum_{a \in A} \tau_a = dM$ [recall that $\sum_{a \in A} \tau_a = \sum_{r=1}^{d} (\alpha_r + \beta_r + \gamma_r) = dB$ by construction]. Since the value of any feasible integral static flow $g \ne g_{\max}$ is at most $d - 1$ and M equals the maximum transit time of a directed s,t-path, a similar calculation shows that the value of a temporally repeated flow induced by an integral static s,t-flow $g \ne g_{\max}$ is at most $dM - B < dM$. Thus, $v_{\max}(T) = dM$ follows.

(ii) Since $g_{\max}$ has to be decomposed into integral path flows and all capacities are one, any path decomposition Γ of $g_{\max}$ will consist of exactly d path flows of value 1 each. Let $g_{\max}$ be decomposed into path flows $g^{(1)}, \ldots, g^{(d)}$ along the directed s,t-paths $P_1, \ldots, P_d$. Each of these paths contains exactly one arc of each of the three sets $\{a_1, \ldots, a_d\}$, $\{a_{d+1}, \ldots, a_{2d}\}$, and $\{a_{2d+1}, \ldots, a_{3d}\}$. Suppose that the paths are indexed such that path P_r consists of the arcs a_r, $a_{d+\phi(r)}$ and $a_{2d+\psi(r)}$, where ϕ and ψ are two permutations of the set $\{1, \ldots, d\}$. The transit time and respectively, the cost of path P_r are given by

$$\tau(P_r) = \alpha_r + \beta_{\phi(r)} + \gamma_{\psi(r)},$$

and, respectively,

$$c(P_r) = 3\tau_{\max} - \alpha_r - \beta_{\phi(r)} - \gamma_{\psi(r)} = 3\tau_{\max} - \tau(P_r).$$

Since we send one unit of flow along each of the paths $P_1, \ldots, P_d$, that is, $|g^{(r)}| = 1$, $r = 1, \ldots, d$, the cost of the induced temporally dynamic flow $f = f^{g,\Gamma,T}$ is given by

$$c^f = \sum_{r=1}^{d} (T + 1 - \tau(P_r)) \cdot c(P_r)$$

$$= \sum_{r=1}^{d} (T + 1 - \tau(P_r)) \cdot (3\tau_{\max} - \tau(P_r))$$

$$= 3d(T + 1)\tau_{\max} - (3\tau_{\max} + T + 1) \sum_{r=1}^{d} \tau(P_r)$$

$$+ \sum_{r=1}^{d} (\tau(P_r))^2 = C_1 + \sum_{r=1}^{d} (\tau(P_r))^2.$$

Since $C = C_1 + dB^2$, it follows that $c^f \leq C$ if and only if $\Sigma_{r=1}^{d} (\tau(P_r))^2 \leq dB^2$, that is, if and only if

$$\tau(P_1) = \cdots = \tau(P_d) = B. \tag{6}$$

(iii) Finally, observe that a set of s,t-paths $P_1, \ldots, P_d$ satisfying (6) corresponds to a solution of the instance I_P of 3-PARTITION since the paths $P_1, \ldots, P_d$ induce two permutations ϕ and ψ of $\{1, \ldots, d\}$ such that $\alpha_r + \beta_{\phi(r)} + \gamma_{\psi(r)} = B$ for all $r = 1, \ldots, d$. ∎

We close this section with the following two remarks:

Remark 3.2. *The construction in the proof of Theorem* 3.2 *above can be used to prove that finding a quickest temporally repeated flow with minimum cost is NP-hard as well. Instead of fixing the time bound T, we fix the flow value $v = dM$, which implies that $T_{\min}(v) = M + B - 1$. Hence, the proof above applies without any further modifications.*

Remark 3.3. *It is easy to see that the MCMTRFP can be solved by the integer program give below. Let $\mathcal{P}$ denote the set of directed s,t-paths. We associate a variable z_P* with each path $P \in \mathcal{P}$, *where z_P is the amount of flow which is sent along path P. We then arrive at the following IP*:

$$\min \quad \sum_{P \in \mathcal{P}} (T + 1 - \tau(P)) \cdot c(P) \cdot z_P \tag{7}$$

$$\text{s.t.} \quad \sum_{P \in \mathcal{P}} (T + 1 - \tau(P)) \cdot z_P \geq v \tag{8}$$

$$\sum_{P \in \mathcal{P}: a \in P} z_P \leq u_a \quad \text{for all } a \in A \tag{9}$$

$$z_P \in {}_0 \quad \text{for all } P \in \mathcal{P}. \tag{10}$$

If the integrality requirement on the path flow variables z_P is dropped and G contains only a polynomial number of directed s,t-paths, the integer program IP turns into a linear program which can be solved in polynomial time in the size of G.

4. A GREEDY APPROACH TO THE MCDFP

In this section, we first propose a greedy algorithm for the MCDFP and then give an exact characterization of the class of graphs for which this greedy algorithm always succeeds. Algorithm GREEDY consists of the following two phases:

1. Determine the set $\mathcal{P} = \{P_1, \ldots, P_K\}$ of all directed s,t-paths in G with transit time $\leq T$ and number the paths within $\mathcal{P}$ such that $c(P_1) \leq c(P_2) \leq \cdots \leq c(P_K)$. Set $r := 1$ and $\tilde{v} := 0$.
2. Repeat the following step as long as $\tilde{v} > v$: At each time $\theta = 0, \ldots, T - \tau(P_r)$, send as much flow as possible along path P_r. Update the flow value $\tilde{v}$ and set $r := r + 1$.

Remark 4.1. *For the special case $c_a = \tau_a$ for all $a \in A$, the above greedy algorithm and the earliest arrival algorithm obtained independently by Minieka* [13] *and Wilkinson* [18] *are similar in spirit. The main difference is that our algorithm uses only directed paths, that is, only forward arcs to augment flow, while the earliest arrival flow algorithm uses also backward arcs.*

Remark 4.2. *Instead of computing the set $\mathcal{P}$ in advance, it is sufficient to compute the paths one by one, sorted increasingly by their cost. However, this observation does not improve the worst-case running time of algorithm GREEDY.*

Obviously, algorithm GREEDY is, in general, not a polynomial time algorithm since its running time depends on the cardinality of the set $\mathcal{P}$. More specifically, the running time of algorithm GREEDY can be bounded as follows:

Lemma 4.1. *Let $\mathcal{P}' = \{P_1, \ldots, P_{K'}\}$, $K' \leq K$, be the set of directed s,t-paths used by algorithm* GREEDY *to build up a feasible dynamic flow of value v and let* TIME($\mathcal{P}'$) *denote the time needed for computing the set $\mathcal{P}'$. Furthermore, let L denote the maximum number of arcs of a path in $\mathcal{P}'$. Then, the overall running time of algorithm* GREEDY *is* $O(\text{TIME}(\mathcal{P}') + LK^2)$.

Proof. In general, the amount of flow sent along an s,t-path $P \in \mathcal{P}'$ will vary with the time θ. However, it is easy to see that the flow on any arc $a \in A$ changes its value over the time interval $[0, T]$ at most $2K' + 1$ times. [Note that each path can add at most two new pieces/intervals to the function $f_a(\tau)$.] Hence, the amount of flow to be augmented along a path $P \in \mathcal{P}'$ can be computed in $O(LK')$ time [$O(K')$ time is needed per arc of the path]. Summing over all paths and noting that $K' \le K$ gives the claimed time bound. ∎

Algorithm GREEDY is closely related to the greedy algorithm of Bein et al. [3] for solving the static minimum-cost flow problem on two-terminal series-parallel networks. The algorithm in [3] is an augmenting path algorithm which successively sends flow along the minimum-cost augmenting path which does not contain any backward arcs.

Observation 4.2. *Algorithm* GREEDY *can be viewed as compact realization of the the greedy algorithm of Bein et al.* [3] *applied to time-expanded graph G*(*T*) *after the removal of the holdover arcs*.

In other words, the dynamic flow obtained by algorithm GREEDY belongs to the class of flows which can be obtained by applying the algorithm of [3] to the time-expanded $G(T)$ without holdover arcs. The advantage of algorithm GREEDY is that it works on the static graph G and not on the typically much larger time expanded graph $G(T)$.

The correctness of Observation 4.2 follows easily from the lemma below which states that, under the assumptions made in this paper, the holdover arcs may be removed from $G(T)$ without affecting the optimal solution of MCDFP:

Lemma 4.3. Fleischer and Skutella [6].* *If all arcs* $a \in A_0 \backslash A$, *that is*, *the arcs which model holdover arcs*, *have nonnegative cost*, *there always exists a minimum-cost dynamic flow which does not use holdover arcs*.

Since we assumed that $c_a \ge 0$ for all $a \in A_0 \backslash A$, the lemma allows us to assume, henceforth, that $A_0 = A$, that is, there are no holdover arcs.

Bein et al. [3] characterized the class of graphs for which their greedy algorithms always determines an optimal flow in the following way:

Proposition 4.4 (Bein et al. [3]). *Let G be an acyclic directed graph*. *The greedy algorithm in* [3] *solves the static minimum-cost flow problem for any choice of the problem parameters* (*arc capacities*, *arc costs*, *and flow value*) *if and only if G is two-terminal series-parallel*.

It is easy to see that the time-expanded graph $G(T)$ of a two-terminal series-parallel graph G is, in general, not two-terminal series-parallel. Hence, Proposition 4.4 cannot be used to obtain a characterization of the class of graphs for which algorithm GREEDY determines a minimum-cost dynamic flow. It will be our next aim to obtain such a characterization. To that end, the following definitions are needed:

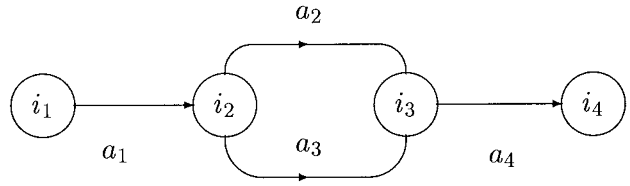

FIG. 2. The rolling-pin graph G_R.

A graph G is called *greedy* if and only if the algorithm GREEDY determines a minimum-cost dynamic flow in G for any choice of the problem parameters (arc capacities u_a, arc costs c_a, arc transit times τ_a, flow value v, and time bound T). The graph G_R depicted in Figure 2 is called the *rolling-pin* graph. The *primitive cycle* graph G_C results from G_R by reversing arc a_3 (cf. Fig. 3).

We start with some easy-to-prove properties of the class of greedy graphs:

Observation 4.5. *G is greedy if and only if each of its subgraphs is greedy. G is greedy if and only if any graph which results from G by subdividing arcs is greedy.*

Observation 4.6. *Graphs G which contain a subgraph that is homeomorphic to* (*i*) *the rolling-pin* G_R *or to* (*ii*) *a directed cycle or to* (*iii*) G_F (*cf. Fig.* 1) *are not greedy.*

Proof. (i) By Observation 4.5, it suffices to show that the graph G_R is not greedy. Let $s = i_1$, $t = i_4$ and set $\tau_{a_2} = 1$, $c_{a_3} = 1$ and all other transit times and costs to zero and all capacities to one. Let $T = 1$ and $v = 2$. GREEDY first selects the two directed s,t-paths $P_1 = (s, a_1, i_2, a_2, i_3, a_4, t)$ and $P_2 = (s, a_1, i_2, a_3, i_3, a_4, t)$ with $c(P_1) = 0$, $c(P_2) = 1$, $\tau(P_1) = 1$, and $\tau(P_2) = 0$. Next, GREEDY sends one unit of flow along path P_1 at time 0, but then it gets stuck because P_2 cannot be used to increase the flow value to $v = 2$.

(ii) Next, assume that G contains a homeomorphic directed cycle Q as subgraph. This yields that G contains a subgraph homeomorphic to the primitive cycle graph G_C defined above (since Q must be reachable via a directed path from s and since t must be reachable via a directed path from Q). Again, it suffices to show that G_C is not greedy. Let $s = i_1$, $t = i_4$ and set $\tau_{a_3} = 1$, $c_{a_3} = -1$, and $c_a = \tau_a = 0$ for all arcs $a \neq a_3$. Let $u_a = 1$ for all arcs a and

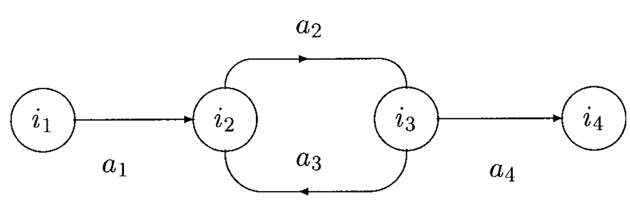

FIG. 3. The primitive cycle graph G_C.

* This result has been stated already in [12], but no proof was provided due to space restrictions and due the fact that the proof we had in mind was quite simple. It turned out, however, that this proof contained a flaw. A correct proof has been provided for a more general case than stated here by Fleischer and Skutella [6]. The results of our paper can be obtained without this lemma, but in a less elegant way, which involves additional case distinctions in the proof of Theorem 4.1.

choose again $T = 1$ and $v = 2$. GREEDY fails for G_C since it starts with path $P_1 = (s, a_1, i_2, a_2, i_3, a_3, i_2, a_2, i_3, a_4, t)$ with $c(P_1) = -1$ and $\tau(P_1) = 1$ and then gets stuck as above.

(iii) Because of (ii), we know that every greedy graph G is acyclic. Hence, the claim follows from Proposition 4.4. (For the graph G_F, the greedy approach already fails in the static case.) ∎

The following theorem presents the main result of this paper, a forbidden subgraph characterization of the class of greedy graphs:

Theorem 4.1. *A graph G is greedy if and only if G is two-terminal series-parallel and does not contain a subgraph homeomorphic to the rolling-pin graph G_R.*

Proof.

"$\Rightarrow$": Follows from Observation 4.6.

"$\Leftarrow$": Suppose that the graph G is not greedy. Then, there must exist an instance I of MCDFP for which algorithm GREEDY fails. Due to Observation 4.2 and Lemma 4.3, this means that, for instance I, at least once there does not exist a minimum-cost augmenting path in the time-expanded graph $G(T)$ using forward arcs only.

Let f' be the dynamic flow obtained by GREEDY immediately before such a situation arises for the first time and let P be a minimum-cost augmenting path in $G(T)$ from s' to t' with respect to the flow f'. We furthermore assume that P has the additional property that it contains a minimum number of blocks of backward arcs, where two backward arcs a and a' of a path P' are said to belong to the same *block* if all arcs of P' which lie between a and a' are backward arcs. Since augmenting cycles with zero cost can be omitted and augmenting cycles with negative cost contradict the optimality of f', we may assume w.l.o.g. that P is a simple path.

Let B be the last block of backward arcs encountered when traversing the path P from s' to t' and denote the first arc of B by a_1 and the last arc of B by a_2. Furthermore, let arc a_3 with tail u and head w be the immediate predecessor of arc a_1 along P and let a_4 with tail v and head x be the immediate successor of arc a_2. Obviously, both a_3 and a_4 are forward arcs.

Let i, j, k, and ℓ be the nodes in G which correspond to the nodes u, v, w, and x in $G(T)$. Then, the following three properties hold:

(a) $i \neq k$, $j \neq k$ and $k \neq \ell$,
(b) $j \neq s$ and $k \neq t$,
(c) G contains a unique directed path from j to k.

Property (a) follows directly from the acyclicity of G and the fact that there are no holdover arcs. To prove property (b), suppose the contrary. First, assume that $j = s$, that is, v is a copy of the source s. Let P' be the subpath of path P from s' to v. Since the arc (s', v) is an artificial arc in $G(T)$ with zero cost and infinite capacity and P' is a minimum-cost path from s' to v, it follows that $c(P') \leq 0$. Consider the cycle Q formed by P' and the (backward) arc (s', v). This cycle is an augmenting cycle since the flow f' sends a positive amount of flow along the arc (s', v) [note that the backward arc a_2 has tail v and (s', v) is the only arc entering v]. Due to the optimality of f', the cost of Q must be 0, which implies that $c(P') = 0$. Let $\tilde{P}$ be the path obtained from P by replacing the subpath P' by the arc (s', v). Since $c(P) = c(\tilde{P})$ holds by construction, $\tilde{P}$ is again a minimum-cost augmenting path, but it contains a smaller number of blocks of backward arcs than does P (actually, it contains no backward arcs at all), which leads to a contradiction. In an analogous way, it can be shown that the assumption $k = t$ leads to a contradiction as well.

To prove property (c), suppose that G contains two distinct directed paths from j to k. Since $j \neq s$ and $k \neq t$, this implies that G contains a homeomorphic rolling-pin (since no arc enters s and no arc leaves t, but j must be reachable by a directed path from s and k must be reachable by a directed path from j) and leads to an immediate contradiction.

Thus, we may, henceforth, assume that $j \neq s$, $k \neq t$ and that there is a single directed path in G from j to k. Since G is acyclic, the following relation $\prec$ defines a partial order on the node set N:

$$i' \prec j' :\Leftrightarrow G \text{ contains a directed path from } i' \text{ to } j'.$$

By the above assumptions, it follows that the nodes i, j, k, and ℓ are pairwise distinct and are ordered such that $i \prec k$, $j \prec k$, and $j \prec \ell$. Furthermore, neither $j \prec i$ nor $\ell \prec k$ can hold since this would imply that the path P contains a cycle or that G contains two directed paths from j to k. It remains to distinguish the following four cases:

Case 1. $i \prec j$ and $k \prec \ell$. It is easy to check that, in this case, there exist directed paths from i to j and k, from j to k and ℓ, and from k to ℓ, such that these five paths have no inner nodes in common. Thus, G contains a subgraph homeomorphic to G_F (set $i_1 := i$, $i_2 := j$, $i_3 := k$, and $i_4 := \ell$), which contradicts the assumption that G is two-terminal series-parallel.

Case 2. $i \nprec j$ and $k \nprec \ell$. Let P_1 be a directed path in G from s to i and P_2 be a directed path from s to j. Since $i \nprec j$, there exists a node $i' \neq i, j$ such that the subpath P'_1 of P_1 from i' to i and the subpath P'_2 of P_2 from i' to j have no inner nodes in common. Similarly, there exists a node $\ell' \neq k, \ell$ and paths P'_3 from ℓ to ℓ' and P'_4 from k to ℓ' which have no inner node in common. It can easily be checked that this implies that G contains a subgraph homeomorphic to G_F (set $i_1 := i'$, $i_2 := j$, $i_3 := k$, and $i_4 := \ell'$), which again leads to a contradiction.

In the remaining two cases, Case 3, where $i \prec j$ and $k \nprec \ell$ holds, and Case 4, where $i \nprec j$ and $k \prec \ell$ holds, a subgraph homeomorphic to G_F and, hence, a contradiction can be obtained in an analogous way by combining the arguments in Cases 1 and 2. ∎

Note that, while the time expanded graph $G(T)$ of greedy graphs G is, in general, not two-terminal series-parallel, it is "almost" two-terminal series-parallel in the following sense:

Theorem 4.2. *Let G be a two-terminal series-parallel graph which does not contain a homeomorphic rolling-pin and let H be any subgraph of the time-expanded graph $G(T)$ which is homeomorphic to the graph G_F. Denote the four essential nodes of H, that is, those nodes which cannot be eliminated by contracting directed paths to single arcs, by u, v, w, and x in such a way that u corresponds to node i_1 of G_F, v to i_2, w to i_3, and x to i_4. Then, at least one of the following two properties is fulfilled*: (*ii*)

(*i*) *u equals the super source s' of $G(T)$ and v is a copy of the source s.*
(*ii*) *x equals the super sink t' of $G(T)$ and w is a copy of the sink t.*

Proof. The proof is similar to the proof of Theorem 4.1, but simpler. We may henceforth assume that G is acyclic since, otherwise, G is not two-terminal series-parallel. Hence, it follows that v and w are the copies of two distinct nodes of G, say i and j. (Recall that we may assume that there no holdover arcs.) We now distinguish two cases:

CASE 1. There exist two distinct directed paths in G from i to j and $i \neq s$ and $j \neq t$. This implies the existence of a homeomorphic rolling-pin in G.

CASE 2. Either $i = s$ or $j = t$ or there exists a unique directed path in G from i to j. Then, either G is not two-terminal series-parallel (in this case, we are again finished) or at least one of the properties (i) and (ii) above is fulfilled. ∎

A consequence of Theorem 4.1 is that the number of directed s,t-paths in a greedy graph G is polynomial in the number of nodes and arcs of G. Hence, the first step of algorithm GREEDY can be performed in polynomial time. Since greedy graphs are acyclic, any directed path in a greedy graph with n nodes has at most $n - 1$ arcs. Using Lemma 4.1 immediately yields the following corollary:

Corollary 4.3. *For a greedy graph G, algorithm* GREEDY *solves the* MCDFP *on G in polynomial time.*

Theorem 4.1 gives a characterization of greedy graphs in terms of forbidden subgraphs. In the following, we will present an alternative characterization which describes how greedy graphs can be built up from certain primitives by series and parallel compositions. To that end, the following property of greedy graphs, which is an immediate consequence of Theorem 4.1, turns out to be essential:

Observation 4.7. *The parallel composition of two greedy graphs is greedy again. Furthermore, if G is greedy, then also the reverse graph G^- is greedy (where G^- results from G by exchanging the role of the source s and the sink t and reversing the direction of each arc of G).*

In the alternative characterization of the class of greedy graphs, the following classes of graphs play a key role:

- A graph G is called a *pumpkin* if it is the parallel composition of $h \geq 1$ directed paths. G is called a *balloon* if it is a series composition of the form $G = G_1 \circ G_2$, where G_1 is a pumpkin and G_2 is a directed path.
- The class of *Type 1 graphs* contains (i) all pumpkins, (ii) all graphs $G = G_1 \circ G_2$, where G_1 is a Type 1 graph and G_2 a directed path, and (iii) all graphs $G = G_1 \parallel G_2$, where G_1 and G_2 are Type 1 graphs.
- The class of *Type 2 graphs* contains all graphs $G = G_1 \circ G_2$, where G_1 is a parallel composition of an arbitrary number of balloons and G_2 is a pumpkin.

Theorem 4.4. *A graph G is greedy if and only if either G or its reverse graph G^- can be represented as a parallel composition of an arbitrary number of Type* 1 *graphs and Type* 2 *graphs.*

An important subclass of the class of greedy graphs is the class of *augmented trees* which are obtained from an in-tree with root t (i.e., a directed tree where all arcs are directed toward the root) by adding a source s and linking s to all leaves of the in-tree.

5. DISCUSSION

In this paper, we proved that the MCDFP is NP-hard for general two-terminal series-parallel graphs and identified a subclass of the class of two-terminal series-parallel graphs for which the problem can be solved in polynomial time by a greedy algorithm. It remains to narrow the gap between the classes of hard and efficiently solvable instances of the MCDFP on two-terminal series-parallel graphs.

Another open problem concerns MCDFP with a *fixed network topology*. In this variant, the underlying network $G = (N, A)$ is fixed *a priori* and the input consists only of the capacities, costs, transit times, flow value, and time bound. Even for this very restricted version of MCDFP, a polynomial time solution is currently out of sight.

Acknowledgments

The authors would like to thank Bruce Hoppe for useful discussions in the early stage of this research and an anonymous referee for a very careful reading of our manuscript and helpful comments.

REFERENCES

[1] J.E. Aronson, A survey on dynamic network flows, Ann Oper Res 20 (1989), 1–66.

[2] J.E. Aronson and B.D. Chen, A forward network simplex algorithm for solving multiperiod network flow problems, Naval Res Log Q 33 (1986), 445–467.

[3] W.W. Bein, P. Brucker, and A. Tamir, Minimum cost flow algorithms for series-parallel networks, Discr Appl Math 10 (1985), 117–124.

[4] R.E. Burkard, K. Dlaska, and B. Klinz, The quickest flow problem, ZOR Methods Models Oper Res 37 (1993), 31–58.

[5] L. Fleischer and M. Skutella, The quickest multicommodity flow problem, Proc 9th Conf on Integer Programming and Combinatorial Optimization, Lecture notes in computer science 2337, Springer, Berlin, 2002, pp. 36–53.

[6] L. Fleischer and M. Skutella, Minimum cost flows over time without intermediate storage, Proc 14th Ann ACM–SIAM Symp on Discrete Algorithms, 2003, pp. 66–75.

[7] L.R. Ford and D.R. Fulkerson, Constructing maximal dynamic flows from static flows, Oper Res 6 (1958), 419–433.

[8] M.R. Garey and D.S. Johnson, Computers and intractability, A guide to the theory of NP-completeness, Freeman, San Francisco, 1979.

[9] A. Hall, S. Hippler, and M. Skutella, Multicommodity flows over time: Efficient algorithms and complexity, Proc 30-th Int Colloq on Automata, Languages and Programming, Lecture notes in computer science, Springer, Berlin, 2003.

[10] B. Hoppe and É. Tardos, Polynomial time algorithms for some evacuation problems, Proc 5-th Ann ACM–SIAM Symp on Discrete Algorithms, 1994, pp. 433–441.

[11] B. Hoppe and É. Tardos, The quickest transshipment problem, Math Oper Res 25 (2000), 36–62; a preliminary version appeared in Proc 6-th Ann ACM–SIAM Symp on Discrete Algorithms, 1995, pp. 512–521.

[12] B. Klinz and G.J. Woeginger, "Minimum cost dynamic flows: The series-parallel case," Integer programming and combinatorial optimization, Proc 4th Int IPCO Conf, Springer lecture notes in computer science 920, E. Balas and J. Clausen (Editors), Springer-Verlag, Berlin, Heidelberg, 1995, pp. 329–343.

[13] E. Minieka, Maximal, lexicographic, and dynamic network flows, Oper Res 21 (1973), 517–527.

[14] J.B. Orlin, Minimum convex cost dynamic network flows, Math Oper Res 9 (1984), 190–207.

[15] C.H. Papadimitriou, P. Serafini, and M. Yannakakis, Computing the throughput of a network with dedicated lines, Discr Appl Math 42 (1993), 271–278.

[16] W.B. Powell, P. Jaillet, and A. Odoni, "Stochastic and dynamic networks and routing," Handbooks in operations research and management science: Network routing, M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser (Editors), Elsevier, North-Holland, Amsterdam, 1995, Vol. 8, pp. 141–295.

[17] J. Valdes, R.E. Tarjan, and E.L. Lawler, The recognition of series-parallel digraphs, SIAM J Comput 11 (1982), 298–313.

[18] W.L. Wilkinson, An algorithm for universal maximal dynamic flows in a network, Oper Res 19 (1971), 1602–1612.