The Maximum Energy-Constrained Dynamic Flow Problem

Sándor P. Fekete[1], Alexander Hall[2], Ekkehard Köhler[3], and Alexander Kröller[1]

[1] Algorithms Group, Braunschweig Institute of Technology, D-38106 Braunschweig, Germany
{s.fekete,a.kroeller}@tu-bs.de
[2] EECS Department, UC Berkeley, CA 94720, USA
alex.hall@gmail.com
[3] Mathematical Institute, Brandenburg University of Technology, D-03013 Cottbus, Germany
ekoehler@math.tu-cottbus.de

Abstract. We study a natural class of flow problems that occur in the context of wireless networks; the objective is to maximize the flow from a set of sources to one sink node within a given time limit, while satisfying a number of constraints. These restrictions include capacities and transit times for edges; in addition, every node has a bound on the amount of transmission it can perform, due to limited battery energy it carries. We show that this *Maximum energy-constrained dynamic flow problem* (ECDF) is difficult in various ways: it is NP-hard for arbitrary transit times; a solution using flow paths can have exponential-size growth; a solution using edge flow values may not exist; and finding an integral solution is NP-hard. On the positive side, we show that the problem can be solved polynomially for uniform transit times for a limited time limit; we give an FPTAS for finding a fractional flow; and, most notably, there is a distributed FPTAS that can be run directly on the network.

1 Introduction

Energy-Constrained Flows and Our Results: Optimizing the flow in a network is one of the classical problems in the theory and practice of algorithms. Even in the early days, more than 50 years ago, it was recognized by Ford and Fulkerson that flow has a temporal dimension, as the flowing objects need time to get to their destination. This is particularly relevant in many current real-world applications, like traffic or communication, where the aspect of time becomes highly important because there is significant variance in the amount of flow over time, so that stationary flows fail to describe the phenomena that are particularly relevant, like congestion. In recent years, a variety of excellent papers has addressed these problems, making tremendous progress in theory, and providing tools for dealing with the challenges of practical adaptive traffic control.

In this paper we describe an additional aspect that becomes important in the context of another highly relevant topic that has attracted a large amount of attention: when considering the flow of information in a distributed and wireless network, one of the essential features is that nodes have limited energy supply. This limits the amount of information they can transmit to their neighbors before dying due to an exhausted battery; as a consequence, routing algorithms do not just have to deal with edge capacities and transit times, but also with limits on the capabilities of nodes to pass on information.

J. Gudmundsson (Ed.): SWAT 2008, LNCS 5124, pp. 114–126, 2008.

We give the first algorithmic study of the resulting *Maximum energy-constrained dynamic flow problem* (ECDF). Our results are as follows.

- With arbitrary transit times, ECDF is NP-hard (Theorem 1). A solution using flow paths can have exponential size growth (Theorem 2). A solution using edge flow values does not seem to exist.
- Finding integral solutions is NP-hard (Theorem 3).
- For uniform transit times and a polynomially bounded time horizon, we show that the problem can be solved in polynomial time (Lemma 1).
- The complexity of finding optimal fractional solutions is an open problem. The problem admits a FPTAS (Theorem 4).
- There is a distributed FPTAS that can be run directly in the network (Theorem 6).

In the remainder of this section, we discuss related work, both on dynamic network flows and on wireless sensor networks. Section 2 provides formal definitions, while Section 3 shows complexity results. Section 4 focuses on the case of uniform transit times; we show that for polynomially bounded time horizon, there is a polynomial-time algorithm, and develop an FPTAS for general time horizon. In Section 5 we finally give a distributed FPTAS, i.e., a class of algorithms that can be run directly on the network.

Flows over time: Already Ford and Fulkerson [11,12] proposed the *dynamic flows* model (also referred to as flows over time), where the individual edges have transit times, determined by the speed at which flow traverses them. Flow rates into edges may vary over time and are bounded by the given capacities. Ford and Fulkerson show that the *maximum s-t-flow over time problem* can be solved in polynomial time.

The *quickest s-t-flow problem* (here a demand is fixed and the goal is to minimize the time horizon T) can be solved by performing a binary search with respect to T, solving a maximum s-t-flow over time problem in each step. For an integral demand Fleischer and Tardos [10] show that the binary search can be stopped after a polynomially bounded number of steps, yielding the optimal time horizon. A much quicker, strongly polynomial algorithm was given by Burkard et al. [3].

In the *quickest transshipment problem* many sources and sinks with supplies and demands are given. Flow can be routed from any source to any sink. This problem appears to be considerably harder than the quickest s-t-flow problem. Nevertheless, Hoppe and Tardos [15,16] manage to give a polynomial time algorithm, which, however, applies submodular function minimization as a subroutine, rendering the algorithm impractical.

Klinz and Woeginger [17] show that computing a *min-cost quickest s-t-flow* in a network with cost coefficients on the edges is already NP-hard for series-parallel graphs. It is even strongly NP-hard to calculate an optimal temporally repeated flow in presence of costs. For the multicommodity version of this problem Fleischer and Skutella [9] propose a $(2 + \varepsilon)$-approximation algorithm.

For two variations of the multicommodity quickest flow over time problem (without costs) Fleischer and Skutella [9] give an FPTAS which is based on condensed time-expanded graphs. Proofs of NP-hardness and strong NP-hardness for these variations are presented in [14]. The NP-hardness proof inspired our proof of Theorem 1.

The survey articles by Aronson [1] and Powell et al. [22] as well as the book published by Ran and Boyce [23] contain examples and references pertinent to the area of flows over time.

Wireless Sensor Networks: In a classical network setting, there is one central authority that knows the full structure of the network, performs all necessary computations, and makes sure that the results are applied throughout the network. It is clear that such an approach is not without its problems; e.g., see [8] for a discussion in the context of traffic. In recent years, there has been growing interest in *distributed algorithms* that lack such a central authority; see [21] for an introduction.

A particular area that has lead to growing interest in distributed algorithms is the field of *wireless sensor networks:* The nodes in the network have limited knowledge of the environment, local communication in a limited neighborhood, no access to a central authority, limited computing and storage capabilities, and limited energy supply, without any chance of getting recharged. The objective is to allow the network to carry out a variety of tasks, using self-organizing and distributed methods. For a current survey of algorithmic models, see [25], and [26] for a discussion of challenges in distributed computing with respect to sensor networks. For an overview of some of our own related algorithmic work, see [6,7,18].

In the WSN context, a problem similar to ECDF has been studied. The *Maximum Lifetime Routing* problem asks for a flow for given demands that maximizes the time until the first node dies. The motivation is that certain nodes collect sensor data and continuously stream them to one or more base stations. All work on this problem models it as following: "maximize T such that there exists a static flow for given demands, where the flow consumes no more than a $1/T$ fraction of each battery."

Madan et al. consider flows with one sink [19,20], using localized subgradient algorithms. Sankar and Liu [24] solve a multi-commodity flow variant using an exponential penalty function. A combinatorial flow augmentation scheme was proposed by Chang and Tassiulas [5]. Zussman and Segall [27] studied a variant with special relay stations between network and data sink. A comparison of practical protocols can be found in Busse et al. [4].

Simply considering static flows that consume just $1/T$ of each battery leaves a gap of up to $\Theta(n)$ to dynamic flows: consider a network of n nodes, connected in a line, with source and sink at the ends, and $T = n - 1$. Our dynamic flow approach exploits the fact that the source-sink-path can be used exactly once, whereas repeating a static flow would just allocate $1/(n-1)$ of the available battery capacities.

Bodlaender et al. [2] consider the maximization of data flowing from the sensors to the data sink, with battery constraints. They neither have an explicit notion of time, nor do they employ fairness constraints that could turn static solutions into dynamic ones.

2 Problem and Definitions

An instance of the ECDF problem comprises the following: The network $G = (V, E)$ with designated source $s \in V$ and sink $t \in V \setminus \{s\}$ (For the multi-source case, see Section 6). We assume that time is sliced into communication rounds. In each round, a node can forward the data it received in the previous round. We believe this is a sufficiently close approximation of the real timing characteristics of store-and-forward wireless mesh networks. Nodes other than the sink can not store flow to be sent later—sensor nodes are usually extremely limited in memory.

Each edge $e \in E$ models a communication link with *bandwidth* $u_e > 0$, which defines the maximum amount of data that can be sent over e in a single communication round. Data can be sent in both directions in a round, the bandwidth applies to the sum.

Each node $v \in V$ has a non-rechargeable battery with capacity $C_v > 0$. There is an *energy consumption model* $c = (c^{\mathrm{s}}, c^{\mathrm{r}})$, with non-negative functions $c^{\mathrm{s}} \in \mathbb{R}^E$, $c^{\mathrm{r}} \in \mathbb{R}^E$. Sending one unit of flow over edge uv, from u to v, decreases u's battery capacity by c^{s}_{uv} and v's capacity by c^{r}_{uv}. We allow for $C_v = \infty$, $c^{\mathrm{s}}_e = 0$, and $c^{\mathrm{r}}_e = 0$; all situations where this leads to divisions by zero or the appearance of ∞ in linear programs are trivial to resolve. The energy model with $c^{\mathrm{s}}_e = 1, c^{\mathrm{r}}_e = 0$ for all $e \in E$ is called the *trivial* model. In the trivial model, the battery capacities become upper bounds for the total flow that can be routed through a node until it dies. The applicability of our results for more realistic models is discussed in Section 6.

Each link has a *transit time* $\tau_e \in \mathbb{N}$. Data sent over e in round θ can be forwarded from the destination in round $\theta + \tau_e$. For the WSN scenario, we assume *uniform* transit times, i.e., $\tau_e = 1$ for all e; we call this problem variant the 1-ECDF problem. We believe the problem with non-uniform transit times is interesting in itself.

There is a given *time horizon* $T \in \mathbb{N}$. A clarification is necessary to avoid "± 1 confusions": We follow the notation from [16], where there are rounds $0, \ldots, T$. Each link e can be used in rounds $0, \ldots, T - \tau_e$. This reflects our wireless network scenario: If you have a time horizon of 1, you can send data over a link once. (There is an opposing model stemming from continuous flows, e.g., water flowing through a tube. There, you need a horizon of 2 for a unit-transit link: One round until the first drop of water reaches the destination, another until all the water is through.)

ECDF problem definition: Putting it together, the problem we want to solve is: *Send as much flow as possible from s to t, such that the edge capacities are obeyed, the flow is delivered within the time horizon T, and no node sends any data after its battery is exhausted.*

To give a concise problem definition, we need a little bit of notation: We denote by $\mathcal{P}^{st}$ the set of all feasible, simple s-t-paths in G. Because we are only interested in s-t-paths, we can safely treat paths as sequences of undirected edges. The length $\tau(P)$ of a path is defined as $\sum_{e \in P} \tau_e$, i.e., the number of edges in P with uniform transit times. A path is feasible if $\tau(P) \leqslant T$. Hence, the source can relay data over P in communication rounds 0 to $T - \tau(P)$, $\varrho(P) := T - \tau(P) + 1 \geqslant 1$ denotes the number of times P can be used. Now let $P = (e_1, e_2, \ldots, e_k)$. We denote by $\tau_{e_i}(P)$ the delay after which data travelling P reaches e_i, i.e., $\tau_{e_i}(P) = \sum_{j=1}^{i-1} \tau_{e_j}$. For a node $v \in V$, $c^*_{v,P}$ denotes the energy drained from v when 1 unit of flow is sent over P, i.e., $c^*_{v,P} = c^{\mathrm{r}}_{e_i} + c^{\mathrm{s}}_{e_{i+1}}$ for inner nodes of the path, where $v \in e_i$ and $v \in e_{i+1}$ for some i. For s resp. t, $c^*_{v,P}$ equals $c^{\mathrm{s}}_{e_1}$ resp. $c^{\mathrm{r}}_{e_k}$.

So, formally we state the ECDF problem as follows:

$$\max \quad \sum_{P \in \mathcal{P}^{st}} \sum_{\theta=0}^{T-\tau(P)} x_P(\theta) \tag{1}$$

$$\text{s.t.} \quad \sum_{\substack{P \ni e: \\ 0 \leqslant \theta - \tau_e(P) \leqslant T - \tau(P)}} x_P(\theta - \tau_e(P)) \leqslant u_e \ \forall e \in E, \theta = 0, \ldots, T \tag{2}$$

$$\sum_{P \ni v} \sum_{\theta=0}^{T-\tau(P)} c^*_{v,P} x_P(\theta) \leqslant C_v \forall v \in V \tag{3}$$

$$x_P(\theta) \geqslant 0 \qquad \forall P \in \mathcal{P}^{st}, \theta = 0, \dots, T - \tau(P)\ . \tag{4}$$

In this LP, $x_P(\theta)$ describes the amount of flow that starts traveling along P in round θ. Inequalities (2) model edge capacities, and inequalities (3) describe node battery constraints. Note that both T and $|\mathcal{P}^{st}|$ can be exponential in the problem's encoding size, and this LP consists of more than $T|E|$ constraints and up to $T|\mathcal{P}^{st}|$ variables.

3 Variants and Complexities

In this section, we analyze the complexity structure of ECDF and some variants.

Theorem 1. *ECDF with arbitrary transit times $\tau_e \in \mathbb{N}$ is* NP*-hard.*

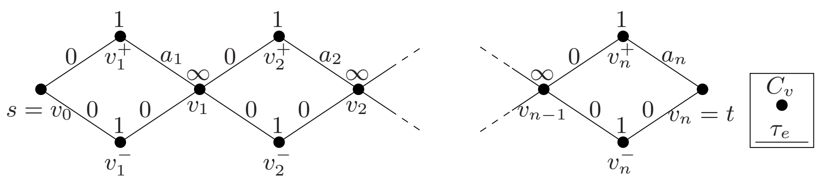

Fig. 1. ECDF instance for the reduction from PARTITION

Proof. Consider an instance of PARTITION: given n positive integers $a_1, \dots, a_n$ with $\sum_{i=1}^n a_i = 2T$ for some $T \in \mathbb{N}$, partition them into two sets of equal weight, i.e., find a $S \subset \{1, \dots, n\}$ such that $\sum_{i \in S} a_i = \sum_{i \notin S} a_i = T$. We claim the PARTITION instance is feasible iff the ECDF instance shown in Figure 1 has an optimal solution of value 2, where the horizon is T, and the energy function is the trivial one; it is obvious that more than 2 is impossible.

It is straightforward to see that a feasible solution $S \subset \{1, \dots, n\}$ for the PARTITION instance induces a feasible pair of paths. For the converse, suppose there is a solution for the ECDF instance that delivers 2 flow units in time. Let $\mathcal{P}$ be the set of flow paths used in the solution, and x_P denote the total flow sent over path $P \in \mathcal{P}$. Because each $\{v_i^+, v_i^-\}$ defines a saturated node cut, $\mathcal{P} \subseteq \{P_S : S \subset \{1, \dots, n\}, \tau(P_S) \leqslant T\}$. Because each v_i^+ is used by some flow paths with total flow 1, we get

$$\sum_{P \in \mathcal{P}} x_P \tau(P) = \sum_{P \in \mathcal{P}} \sum_{i: v_i^+ \in P} x_P a_i = \sum_{i=1}^n a_i \sum_{P \in \mathcal{P}: v_i^+ \in P} x_P = \sum_{i=1}^n a_i = 2T\ .$$

Because $\sum_{P \in \mathcal{P}} x_P = 2$ and $\tau(P) \leqslant T$ for all $P \in \mathcal{P}$, this only holds for $\tau(P) = T$, so every $P_S \in \mathcal{P}$ induces a solution S for the PARTITION instance. $\square$

An important property of static network flows is the existence of two popular encoding schemes with polynomial size: First, edge-based, where there is a flow value for every

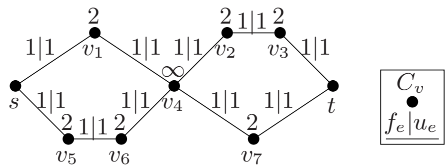

Fig. 2. Network and edge flow values where the path decomposition matters

edge. Second, path-based, where there is a flow value for every path. This even holds for the maximum dynamic flow problem [11]. Now we show that both schemes are not applicable to ECDF with arbitrary transit times. For the edge-based encoding we interpret edge flow values as the maximum flow that is sent over an edge. Consider the network and flow in Figure 2. There are four s-t-paths in it: $P_{\cap\cap} = (s, v_1, v_4, v_2, v_3, t)$, $P_{\cap\cup} = (s, v_1, v_4, v_7, t)$, $P_{\cup\cap} = (s, v_5, v_6, v_4, v_2, v_3, t)$ and $P_{\cup\cup} = (s, v_5, v_6, v_4, v_7, t)$. We show that the given solution—with flow value 1 for every edge—has different values depending on the path decomposition. Let $T = 6$. If we use $P_{\cap\cap}$ and $P_{\cup\cup}$, we can use both paths twice, with a total flow of 4. On the other hand, if we use $P_{\cap\cup}$ and $P_{\cup\cap}$, the paths may be used once resp. twice, giving a total flow of 3. The only edge-based solution encoding that we are aware of assigns time-dependent flow functions $f_e : \{0, \ldots, T\} \to \mathbb{R}$ to the edges, which are not necessarily of polynomial size.

Unfortunately, the usual workaround of using path-based formulation does not help:

Theorem 2. *There are instances for ECDF with arbitrary transit times that allow no optimal solution consisting of a polynomial number of flow paths.*

Proof sketch. The proof employs a sequence of k chained cycles. In each cycle, a flow path can either use a path with transit time 2^{k-1}, or another one with zero transit. Both paths have a flow limit of 2^{k-1}, enforced by batteries. The edge leading to t has a capacity of 1. It can be shown that any optimal flow must deliver one flow unit to t in every time slot from 0 to $T = 2^k - 1$, and each must travel on a different flow path. $\square$

Theorem 3. *Finding an ECDF solution with integral flow values is* NP*-hard.*

Proof. By reduction from the following strongly NP-hard 3-PARTITION variant: given three sets of positive integers $\{a_1, \ldots, a_n\}$, $\{b_1, \ldots, b_n\}$, and $\{c_1, \ldots, c_n\}$ with $L := \sum_{i=1}^n a_i = \sum_{i=1}^n b_i = \sum_{i=1}^n c_i$, find a partition into n triples of equal weight, i.e., find permutations $\alpha, \beta, \gamma \in S_n$ such that $a_{\alpha(i)} + b_{\beta(i)} + c_{\gamma(i)} = 3L/n$ for all i.

We construct an ECDF instance $G = (V, E)$ as follows: For each of the $3n$ numbers, say a_i, there is a chain A_i consisting of a_i nodes connected in line, of which a_i^+ is the first and a_i^- is the last. Additionally, there are the source s and sink t. The source is connected to each entry node of the first set, that is, $sa_i^+ \in E$ for all i. Each exit node of the first set is connected to each entry of the second: $a_i^- b_j^+ \in E$ for all i, j. Analogously, the exits from the second set are linked to the entries of the third, and all exits from the third are linked to t. Each edge $e \in E$ has unit capacity $u_e = 1$. Furthermore, each node $v \neq s, t$ has a battery capacity of $C_v = 1$, where the power consumption model is the trivial one: $c_e^{\mathrm{s}} = 1, c_e^{\mathrm{r}} = 0$ for all $e \in E$. The time horizon is $T = 3L/n + 1$.

We claim that the 3-PARTITION instance is solvable iff the optimal integral ECDF solution value is n. There cannot be a total flow of more than n because $\{a_1^+,\dots,a_n^+\}$ forms a node cut with total energy n.

Assume there is a feasible, integral ECDF solution $(x_P(\theta))_{P\in\mathcal{P},\theta\in\theta(P)}$, where $\mathcal{P}\subseteq\mathcal{P}^{st}$, $\theta(P)\subseteq\{0,\dots,T-\tau(P)\}$, and $x_P(\theta)\in\mathbb{N}^+$ for all $P\in\mathcal{P},\theta\in\theta(P)$. Assume it has value n. Because each of the sets $\{a_1^+,\dots,a_n^+\}$, $\{a_1^-,\dots,a_n^-\}$, $\{b_1^+,\dots,b_n^+\}$, …, $\{c_1^-,\dots,c_n^-\}$ is a saturated node cut, it cannot be crossed twice by any path. Hence, each $P\in\mathcal{P}$ is one of the paths $P_{i,j,k}=(s,A_i,B_j,C_k,t)$, $i,j,k=1,\dots,n$. Because the flow is integral and all battery capacities are 1, each $x_P(\theta)=1$. Each of the chains in the graph can be used by just one flow path due to its battery capacity, and because the total flow is n, each chain is used by exactly one path. So $\mathcal{P}=\{P_{\alpha(i),\beta(i),\gamma(i)}:i=1,\dots,n\}$ for some $\alpha,\beta,\gamma\in S_n$. We know that

$$\sum_{i=1}^{n}\tau(P_{\alpha(i),\beta(i),\gamma(i)})=\sum_{i=1}^{n}(a_{\alpha(i)}+b_{\beta(i)}+c_{\gamma(i)}+1)=\sum_{i=1}^{n}a_i+\sum_{i=1}^{n}b_i+\sum_{i=1}^{n}c_i+n=nT,$$

and, because no path length can exceed T due to the feasibility of the solution, we conclude that each path length must equal $T=3L/n+1$. Therefore $a_{\alpha(i)}+b_{\beta(i)}+c_{\gamma(i)}=3L/n$ for each $i=1,\dots,n$, proving that α,β,γ is feasible for the 3-PARTITION instance. It is straighforward to see the converse. □

The proof does not carry over to the fractional ECDF problem: There is a trivial LP formulation for the ECDF instance that uses the n^3 possible flow paths explicitly.

4 Centralized Algorithms for 1-ECDF

In this section, we concentrate on ECDF with uniform transit times.

Lemma 1. *1-ECDF is polynomial-time solvable, if T is polynomially bounded in n.*

Proof. The time-expanded graph $G(T)$ has polynomial size and thus allows for a simple edge-based LP. □

A temporally repeated ("TR") flow is a flow $(x_P(\theta))_{P,\theta}$ where each path carries the same amount of flow at all times, i.e., $x_P(\theta)=x_P(\theta')$ for all $\theta,\theta'\in\{0,\dots,T-\tau(P)\}$. When $T>2n$, the problem of finding a temporally repeated 1-ECDF solution can be formulated as follows:

$$\max \sum_{P\in\mathcal{P}^{st}}\varrho(P)x_P \tag{5}$$

$$\text{s.t.} \sum_{P\ni e}x_P\leqslant u_e \qquad \forall e\in E \tag{6}$$

$$\sum_{P\ni v}\varrho(P)c^*_{v,P}x_P\leqslant C_v \quad \forall v\in V \tag{7}$$

$$x_P\geqslant 0\,. \tag{8}$$

The restriction $T>2n$ comes from inequality (6) which is only valid if all the paths that use some edge e send their flow over e in at least one common point in time.

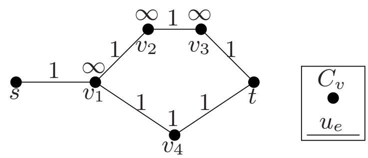

Fig. 3. Network with a gap between optimal and temporally repeated solutions

Lemma 2. *Maximum temporally repeated solutions for 1-ECDF can be found in polynomial time.*

Proof. The dual LP of (5)–(8) is

$$\min \sum_{v \in V} C_v \mu_v + \sum_{e \in E} u_e \pi_e \tag{9}$$

$$\text{s.t.} \sum_{v \in P} \varrho(P) c^*_{v,P} \mu_v + \sum_{e \in P} \pi_e \geqslant \varrho(P) \quad \forall P \in \mathcal{P} \tag{10}$$

$$\mu_v \geqslant 0 \ \forall v, \qquad \pi_e \geqslant 0 \ \forall e \tag{11}$$

The separation problem for this LP is to find a violated inequality (10), given edge weights $(\pi_e)_{e \in E}$ and node weights $(\mu_v)_{v \in V}$: Find a path $P \in \mathcal{P}^{st}$ satisfying

$$\sum_{v \in P} c^*_{v,P} \mu_v + \sum_{e \in P} \frac{1}{\varrho(P)} \pi_e < 1 \tag{12}$$

or prove that no such path exists. The left-hand-side of (12) can be rewritten as

$$\sum_{v \in P} c^*_{v,P} \mu_v + \sum_{e \in P} \frac{1}{\varrho(P)} \pi_e = \sum_{e = uv \in P} \left(\tfrac{1}{T - \tau(P) + 1} \pi_e + c^s_e \mu_u + c^r_e \mu_v \right) \tag{13}$$

which is just the length of P according to some length-dependent edge weights. So the separation problem reduces to the question whether the shortest path in $\mathcal{P}^{st}$ according to this weight function has a length strictly less than 1.

Because $\tau(P) \in \{1, \ldots, n\}$ for each $P \in \mathcal{P}^{st}$, there are just n possible values for $\frac{1}{T - \tau(P) + 1}$. We can find the shortest path by enumerating over these values. In each step, we seek the shortest path consisting of exactly k edges for some $k \in \{1, \ldots, n\}$. This can be done in polynomial time by searching for the shortest path from $s(0)$ to $t(k)$ in the time-expanded graph $G(k)$. □

Lemma 3. *Temporally repeated 1-ECDF solutions are not always optimal.*

Proof. Consider the network shown in Figure 3. The horizon is $T = 4$, communication cost is the trivial one. There are two paths in this network: The "upper" one $P = (s, v_1, v_2, v_3, t)$ that can be used exactly once, and the "lower" one $Q = (s, v_1, v_4, t)$, that can be used twice with a total flow of 1 due to the battery limitation at v_4. An optimal solution sends 1 flow unit along P at time 0 and another unit along Q at time 1, with a total flow value of 2. Optimality holds because edge sv_1 is saturated over time.

A temporally repeated solution sends x_P along P at time 0 and x_Q along Q at times 0 and 1. Because of the capacity of sv_1, $x_P + x_Q \leqslant 1$ holds. Furthermore, $2x_Q \leqslant 1$ due to the battery capacity at v_4. The total flow is $x_P + 2x_Q$, which is maximized by $x_P = x_Q = \frac{1}{2}$ with an objective value of $\frac{3}{2}$. □

Lemma 4. *For $T > \lambda n$, $\lambda \geqslant 2$ the value of a maximum temporally repeated solution* TR *is greater than or equal to* $\frac{\lambda-1}{\lambda}\mathrm{OPT}$, *where* OPT *be the value of an optimal 1-ECDF solution.*

Proof. Let $x = (x_P(\theta))_{P,\theta}$ be an optimal solution. We construct a temporally repeated solution $y = (y_P)_P$ from it by averaging over all path flows.

So let $y_P := \frac{1}{\varrho(P)} \sum_{\theta=0}^{T-\tau(P)} x_P(\theta)$ for each $P \in \mathcal{P}^{st}$. This flow satisfies all battery capacity constraints, because each flow path carries the same total flow as in x, and it delivers the same flow within the horizon. It may violate edge capacities though. So let $e \in E$. Then the load on e at time θ is

$$\sum_{\substack{P \ni e: \\ 0 \leqslant \theta - \tau_e(P)}} y_P \leqslant \sum_{P \ni e} y_P \leqslant \frac{1}{T-n} \sum_{P \ni e} \sum_{\theta=0}^{T-\tau(P)} x_P(\theta) \leqslant \frac{1}{T-n} T u_e \leqslant \frac{\lambda}{\lambda-1} u_e \ .$$

Hence $\frac{\lambda-1}{\lambda} y$ is a feasible temporally repeated flow. □

Theorem 4. *1-ECDF admits a FPTAS.*

Proof. Let $\varepsilon > 0$. If $T \leqslant \frac{1}{\varepsilon} n$ or $T \leqslant 2n$, we can solve the problem by Lemma 1. Otherwise, by Lemma 2 we can compute a maximum temporally repeated flow in polynomial time and, by Lemma 4, its value is at least $((\frac{1}{\varepsilon} - 1)/\frac{1}{\varepsilon})\mathrm{OPT} = (1-\varepsilon)\mathrm{OPT}$. □

5 Distributed Algorithm for 1-ECDF

In this section, we propose a distributed FPTAS for 1-ECDF. The core idea relies on the approximation algorithm [13] for fractional packing problems by Garg and Könemann, so we briefly review their algorithm: Consider a fractional packing LP of the type $\max\{c^{\mathsf{T}} x | Ax \leqslant b, x \geqslant 0\}$ with a $\tilde{m} \times \tilde{n}$-matrix A, where all coefficients in A, b, and c are nonnegative. Its dual is the covering LP $\min\{y^{\mathsf{T}} b | y^{\mathsf{T}} A \geqslant c^{\mathsf{T}}, y \geqslant 0\}$. Initially, $x = 0$ and $y_j = \delta / b_j$ for all $j = 1, \ldots, \tilde{m}$, where $\delta := (1+\varepsilon)((1+\varepsilon)\tilde{m})^{-1/\varepsilon}$. The algorithm repeats the following iteration until $y^{\mathsf{T}} b \geqslant 1$: Let i^* be the primal variable (think "maximally violated" dual constraint) that minimizes $(y^T A)_i / c_i$ for $i \in \{1, \ldots, \tilde{n} | c_i > 0\}$. Let j^* be the primal constraint (think "minimum capacity edge/node") that minimizes $b_j / A_{i^* j}$ for $j = 1, \ldots, \tilde{m}$ where $A_{ij} \neq 0$. Now increase x_{i^*} by $b_{j^*} / A_{i^* j^*}$ (corresponding to "sending flow over i^*"), and update the dual variables: $y_j := y_j (1+\varepsilon)(b_{j^*} / A_{i^* j^*}) / (b_j / A_{i^* j})$ for all $j = 1, \ldots, \tilde{m}$ with $b_j \neq 0$.

Finally, a feasible primal solution can be obtained by scaling down all variables such that all primal constraints are obeyed; a scaling factor of $1/\log_{1+\varepsilon}((1+\varepsilon)/\delta)$ is sufficient. This can also be done during the increase of primal variables, i.e., during routing of flow, if desired.

Theorem 5 (Garg, Könemann [13]). *Using an oracle that finds a Λ-approximation for the maximally violated constraint, the G&K algorithm computes a $\Lambda(1-\varepsilon)^{-2}$-approximation in $\tilde{m}\lceil \frac{1}{\varepsilon}\log_{1+\varepsilon}\tilde{m}\rceil$ iterations.*

Note that [13] only deals with optimal dual separation, i.e., $\Lambda = 1$. The extension for arbitrary $\Lambda > 1$ is straightforward.

Similar to the previous section, we solve 1-ECDF by distinguishing two cases: $T > \frac{1}{\varepsilon}n$, where the TR gap is small, and $T \leqslant \frac{1}{\varepsilon}n$, where the horizon is polynomially bounded. Actually, the G&K algorithm can easily be distributed, given a fast approximation for the dual separation. For this purpose, we show how to reduce the n shortest path computations we needed in the proof of Lemma 2 to one:

Lemma 5. *Let $T > \lambda n$, $\lambda \geqslant 2$. Let $\pi_e \geqslant 0$ for all $e \in E$ and $\mu_v \geqslant 0$ for all $v \in V$. Then the dual separation problem* (12) *for temporally repeated flows can be $\frac{\lambda}{\lambda-1}$-approximated using a single shortest path computation.*

Proof. The separation problem is to find a shortest path according to the length function

$$z(P) := \sum_{uv\in P}\left(\tfrac{1}{\varrho(P)}\pi_{uv} + c^s_{uv}\mu_u + c^r_{uv}\mu_v\right). \tag{14}$$

We define another function to approximate z:

$$y(P) := \sum_{uv\in P}\left(\tfrac{1}{T}\pi_{uv} + \tfrac{\lambda-1}{\lambda}(c^s_{uv}\mu_u + c^r_{uv}\mu_v)\right). \tag{15}$$

Observe that for each $P \in \mathcal{P}^{st}$, $T \geqslant \varrho(P) \geqslant T - n > T - \frac{1}{\lambda}T = \frac{\lambda-1}{\lambda}T$ holds; hence $\frac{1}{T} \leqslant \frac{1}{\varrho(P)} \leqslant \frac{\lambda}{\lambda-1}\frac{1}{T}$. This proves that $y(P) \leqslant z(P) \leqslant \frac{\lambda}{\lambda-1}y(P)$ for every $P \in \mathcal{P}^{st}$. Therefore, the minimum-y path is a $\frac{\lambda}{\lambda-1}$-approximation to the minimum-z path. Now $y(P)$ is just a sum of directed, non-negative edge weights. Then, the minimum-y path can be found by a single run of any shortest path algorithm. $\square$

G&K is turned into distributed algorithm for the large-T case as follows:

Algorithm 1. Distributed algorithm for $T > \frac{1}{\varepsilon}n$

Each node $v \in V$ initializes and stores μ_v and π_e for each $e \in \delta(v)$;
repeat
  s initiates a distributed Bellman-Ford shortest path algorithm;
  The network reports the approximate shortest path's weight and capacity to s;
  The network augments flow along this path, each nodes updates the dual weight, and reports the dual objective increase back to s;
until *s observes that the dual objective is at least* 1 ;
Scale flow for feasibility, unless already done during augmentation.

Lemma 6. *Let $\varepsilon > 0$ and $T > \frac{1}{\varepsilon}n$, $\frac{1}{\varepsilon} \geqslant 2$. Then Algorithm 1 is a $(1-\varepsilon)^{-4}$-approximation for the ECDF problem and runs in time $O(n(n+m)\frac{1}{\varepsilon}\log_{1+\varepsilon}(n+m))$*

Proof. According to Lemma 5, the Bellman-Ford algorithm computes an approximation to the dual separation problem with ratio $\frac{\lambda}{\lambda-1} = (1-\varepsilon)^{-1}$. Hence, Algorithm 1 is a $(1-\varepsilon)^{-3}$-approximation to finding a temporally repeated ECDF solution (Theorem 5). Because a maximum temporally repeated flow is a $(1-\varepsilon)^{-1}$-approximation to the original ECDF problem (Lemma 4), the solution found by Algorithm 1 is a $(1-\varepsilon)^{-4}$-approximation. The runtime results from the iteration bound of Theorem 5 and the $O(n)$ runtime of a distributed Bellman-Ford computation. □

The other case (small T) is mostly analogous, so we just give the result:

Lemma 7. *Let $\varepsilon > 0$ and $T \leqslant \frac{1}{\varepsilon}n$. Then there is a distributed algorithm that finds a $(1-\varepsilon)^{-2}$-approximation in time $O(\frac{1}{\varepsilon^2}mn^2 \log_{1+\varepsilon}(\frac{1}{\varepsilon}mn))$.*

Proof sketch. Run a distributed variant of G&K, similar to Algorithm 1 on the exact LP formulation (1)-(4). □

Together, Lemmas 6 and 7 (and the special case $\varepsilon \geqslant \frac{1}{2}$, $T > \frac{1}{\varepsilon}$, which is trivial to resolve) allow us to state the main theorem of this section:

Theorem 6. *ECDF admits a distributed FPTAS. Each node $v \in V$ needs to store no more than $O(p(v)+1)$ many variables, where $p(v)$ denotes the number of flow paths using v in the solution.*

6 Extensions

Multi-terminal variants: ECDF problems where there is one source and many sinks can be solved by our algorithms, both centralized and distributed, as well. It is sufficient to alter $\mathcal{P}^{st}$ accordingly. The opposite case with one sink and many sources can be solved by exchanging them and reversing time. This just applies to the objective of maximizing the total flow though, as max-min objective variants are no longer fractional packing problems. The situation is similar for multi-commodity settings with many sources and sinks: Maximizing the total flow is possible by adjusting $\mathcal{P}^{st}$—in the distributed setting an additional syncing step between the sources is needed in each iteration. The max-min multi-commodity case can not be solved using our algorithms.

Geometric communication cost functions: In wireless networks, it is a common assumption that the sending cost for transmitting over a link e of length d is $c_e^{\mathrm{s}} = \Theta(d^\alpha)$ for some constant $\alpha \in [2,6]$. The cost for receiving is often modelled as either 0 or $\Theta(c_e^{\mathrm{s}})$. Our constructive results work for any cost function. We have stated the negative results in Section 3 using the trivial cost function for clarity. Note that the problem instances used in the proofs can be embedded such that every link has length 1 (actually, the figures show such embeddings), where the geometric cost function becomes the trivial one. Hence these results apply as well.

7 Conclusion and Open Problems

In this paper, we introduced the novel ECDF problem, which has direct applications in distributed networks, e.g., sensor and ad-hoc network. We proved various negative results and show that an FPTAS exists for the network-motivated 1-ECDF variant, which can be turned into a distributed FPTAS.

There are various related problems of interest. Essentially, we started out with a well-studied problem (maximum dynamic flow) and added two features: firstly, the battery constraints that make the problem much harder (Theorems 1 and 2); and secondly, uniform transit times, making it easier. Because both constraints are important in sensor and ad-hoc networks, studying other dynamic flow problems like quickest flow/transshipment with these extensions poses interesting new challenges.

A tantalizing open problem is the complexity of the fractional ECDF problem with arbitrary transit times. We conjecture that this is in P: consider the path-based LP formulation. Allowing flow changes only in the first and last n steps (i.e., adding $x_P(n) = x_P(n+1) = \ldots = x_P(T-n-1)$ for all P to the formulation) may not change the problem. This new formulation is in P, because the dual separation problem can be easily solved similar to Lemma 2. Alas, we lack a proof.

Another open problem is the existence of an encoding scheme for the solutions with polynomial size (cf. Theorem 2) resp. whether the decision variant of ECDF with arbitrary (or uniform) transit time is in NP.

References

1. Aronson, J.E.: A survey of dynamic network flows. Annals of OR 20, 1–66 (1989)
2. Bodlaender, H., Tan, R., van Dijk, T., van Leeuwen, J.: Integer maximum flow in wireless sensor networks with energy constraint. In: Proc. SWAT (2008)
3. Burkard, R.E., Dlaska, K., Klinz, B.: The quickest flow problem. ZOR — Methods and Models of Operations Research 37, 31–58 (1993)
4. Busse, M., Haenselmann, T., Effelsberg, W.: A comparison of lifetime-efficient forwarding strategies for wireless sensor networks. In: Proc. PE-WASUN, pp. 33–40 (2006)
5. Chang, J.-H., Tassiulas, L.: Maximum lifetime routing in wireless sensor networks. IEEE/ACM Transactions on Networking 12(4), 609–619 (2004)
6. Fekete, S.P., Kröller, A.: Geometry-based reasoning for a large sensor network. In: Proc. SoCG, pp. 475–476 (2006)
7. Fekete, S.P., Kröller, A., Pfisterer, D., Fischer, S.: Algorithmic aspects of large sensor networks. In: Proc MSWSN, pp. 141–152 (2006)
8. Fekete, S.P., Schmidt, C., Wegener, A., Fischer, S.: Hovering data clouds for recognizing traffic jams. In: Proc. IEEE-ISOLA, pp. 213–218 (2006)
9. Fleischer, L., Skutella, M.: Quickest flows over time. SIAM Journal on Computing 36, 1600–1630 (2007)
10. Fleischer, L.K., Tardos, É.: Efficient continuous-time dynamic network flow algorithms. Operations Research Letters 23, 71–80 (1998)
11. Ford, L.R., Fulkerson, D.R.: Constructing maximal dynamic flows from static flows. Operations Research 6, 419–433 (1958)
12. Ford, L.R., Fulkerson, D.R.: Flows in Networks. Princeton University Press, Princeton (1962)
13. Garg, N., Könemann, J.: Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In: Proc. FOCS, p. 300 (1998)
14. Hall, A., Hippler, S., Skutella, M.: Multicommodity flows over time: Efficient algorithms and complexity. Theoretical Computer Science 379, 387–404 (2007)
15. Hoppe, B., Tardos, É.: The quickest transshipment problem. Mathematics of Operations Research 25, 36–62 (2000)
16. Hoppe, B.E.: Efficient dynamic network flow algorithms. PhD thesis, Cornell (1995)

17. Klinz, B., Woeginger, G.J.: Minimum cost dynamic flows: The series-parallel case. In: Balas, E., Clausen, J. (eds.) IPCO 1995. LNCS, vol. 920, pp. 329–343. Springer, Heidelberg (1995)
18. Kröller, A., Fekete, S.P., Pfisterer, D., Fischer, S.: Deterministic boundary recognition and topology extraction for large sensor networks. In: Proc. SODA, pp. 1000–1009 (2006)
19. Madan, R., Lall, S.: Distributed algorithms for maximum lifetime routing in wireless sensor networks. IEEE Transactions on Wireless Communications 5(8), 2185–2193 (2006)
20. Madan, R., Luo, Z.-Q., Lall, S.: A distributed algorithm with linear convergence for maximum lifetime routing in wireless networks. In: Proc. Allerton Conference, pp. 896–905 (2005)
21. Peleg, D.: Distributed computing: a locality-sensitive approach. SIAM, Philadelphia (2000)
22. Powell, W.B., Jaillet, P., Odoni, A.: Stochastic and dynamic networks and routing. In: Network Routing, ch. 3. Handbooks in Operations Research and Management Science, vol. 8, pp. 141–295. North–Holland, Amsterdam, The Netherlands (1995)
23. Ran, B., Boyce, D.E.: Modelling Dynamic Transportation Networks. Springer, Heidelberg (1996)
24. Sankar, A., Liu, Z.: Maximum lifetime routing in wireless ad-hoc networks. In: Proc. INFOCOM, pp. 1089–1097 (2004)
25. Schmid, S., Wattenhofer, R.: Algorithmic models for sensor networks. In: Proc. IPDPS (2006)
26. Wattenhofer, R.: Sensor networks: Distributed algorithms reloaded - or revolutions? In: Flocchini, P., Gąsieniec, L. (eds.) SIROCCO 2006. LNCS, vol. 4056, pp. 24–28. Springer, Heidelberg (2006)
27. Zussman, G., Segall, A.: Energy efficient routing in ad hoc disaster recovery networks. In: Proc. INFOCOM, pp. 682–691 (2003)