Theoretical Computer Science 412 (2011) 391–401

Contents lists available at ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Shortest path and maximum flow problems in networks with additive losses and gains☆,☆☆

Franz J. Brandenburg[a,*], Mao-cheng Cai[b]

[a] University of Passau, 94030 Passau, Germany

[b] Institute of Systems Science, Chinese Academy of Sciences, Beijing 100080, China

ARTICLE INFO

Keywords:
Extended networks
Lossy and gainy arcs
Max-flow problems
Shortest path problems
$\mathcal{NP}$-hard problems
Unit-loss networks

ABSTRACT

We introduce networks with additive losses and gains on the arcs. If a positive flow of x units enters an arc a, then $x + g(a)$ units exit. Arcs may increase or consume flow, i.e., they are gainy or lossy. Such networks have various applications, e.g., in financial analysis, transportation, and data communication.

Problems in such networks are generally intractable. In particular, the shortest path problem is $\mathcal{NP}$-hard. However, there is a pseudo-polynomial time algorithm for the problem with nonnegative costs and gains. The maximum flow problem is strongly $\mathcal{NP}$-hard, even in networks with integral capacities and with unit gain or with loss two on the arcs, and is hard to approximate. However, it is solvable in polynomial time in unit-loss networks using the Edmonds–Karp algorithm.

Our $\mathcal{NP}$-hardness results contrast efficient polynomial time solutions of path and flow problems in standard and in so-called generalized networks with multiplicative losses and gains.

1. Introduction

How much does it cost to transport commodities in a network with losses and gains on the arcs? What is an optimal investment policy when charges and fees, interest and changes of stock quotations are taken into account? How much data can be sent through a communication network if the amount of data is changed by the used links and routers, e.g., by appending routing information? In such networks all nodes except the source and the sink are flow conserving, whereas the arcs are not. A gain function on the arcs augments or decreases the flow. If $x > 0$ units enter an arc a, then $x + g(a)$ units exit, where $g(a)$ is the gain of the arc a in a so-called additive network.

In the general setting there is a gain function g_a for each arc a such that $g_a(x)$ units exit if x units enter a. The amount of flow must be nonnegative at either end, and $g_a(0) = 0$ expresses "no flow, no gain". An arc a is gainy and increases the flow if $g_a(x) - x > 0$, and is lossy if $g_a(x) - x < 0$, where some flow is consumed. Each arc has a capacity and a cost per unit which are taken at the entrance.

Networks with losses and gains have many applications ranging from historical events in trading to current problems in financial analysis and data communication. Losses and gains express the variation of the commodity by the evaporation of gas and liquids, by interest and quotations on stock, or by breeding. With multiplicative losses and gains, i.e., with linear gain functions, the amount of flow on an arc is changed by a certain percentage. The loss or gain is proportional to the size of the transported commodity. Such networks are known as *generalized networks*. They have been studied intensively in

☆ This research was done in part while the authors were visiting the City University of Hong Kong.

☆☆ This paper has been selected for publication in TCS-A among the best papers presented at the Third International Workshop on Frontiers in Algorithmics (FAW 2009), Hefei, China, June 20–23, 2009.

* Corresponding author. Tel.: +49 851 5093030; fax: +49 851 509 3032.
E-mail addresses: brandenb@informatik.uni-passau.de, brandenb@fim.uni-passau.de (F.J. Brandenburg), caimc@mail.iss.ac.cn (M.-c. Cai).

0304-3975/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2010.11.019

the literature from the early days of production planning and combinatorial optimization [17,8,10,19] to recent research [1,4,9,12,25,27]. Here many network flow problems have a canonical linear programming (LP) formulation, which implies that there is a polynomial time solution by general purpose LP-methods [18,21]. Recently, efficient combinatorial algorithms have been developed, see [25] for the shortest path problem, [12–14] for the maximum flow problem, and [27] for the min-cost problem. Oldham [25] provides a summary for the solution of various path and flow problems in generalized networks.

Accordingly, there are many applications with nonlinear and, in particular, with additive gain functions. In communication networks the data is expanded by appending a fixed amount of routing information, which means an additive gain, as described in the debugging mode of routers [6]. In the transport of goods or liquids a fixed amount of a commodity may be lost due to consumption, leakage, theft, or toll. For example, at irrigation systems or rivers that run dry seasonally an initial amount of water seeps away before the waterways are saturated by fresh water. The historic "Raffelstetter Zollordnung" (Raffelstett customs regulations) from 903/05 stated that "three bushels of salt per ship" had to be delivered at the toll gates in Passau, Linz or Mautern when crossing the Danube river, and "one bushel of salt per wagon" on the alternative overland routes [15]. Similar regulations were reported for the caravans from China to Europe transporting tea on the Silk Road and paying toll in terms of tea (http://www.tee-import.de/china/chronolog.htm). The withhold toll was a fixed or a linear amount of the transported commodities, and this loss accumulated with the number of tollgates on the route. The major application of gain functions comes from financial transactions in business. Here linear, additive, and even more complex functions are used to describe the transaction costs of financial transactions, money exchange and stock dealing. Such costs are investigated in financial analysis [24]. There are fees, such as a service charge for currency exchange, a commission for trading investments, and fees for the exchange of stock [28]. These are fixed costs and are a loss for the investor. In addition the investor's capital changes by interest, quotations and exchange rates, which can be expressed by linear gain functions. When all price alterations and fees are taken into account, the optimal investment policy can be modeled as a shortest (cheapest) path problem or as a maximum flow problem in networks with losses and gains. Unfortunately, this works only backwards for the past and not for future investments.

In this paper we introduce and investigate *additive networks*, which extend standard networks by additive gain functions of the form $x + g(a)$ for the arcs. These gain functions are associated with the network and the flow. An arc is used if the flow is positive and does not exceed the capacity at the entrance and is nonnegative at the exit. Then the flow is increased or decreased by a fixed amount. In detail, a flow x on an arc a with capacity $u(a)$ and gain $g(a)$ becomes $x + g(a)$ if $x > 0$, $x \leq u(a)$ and $x + g(x) \geq 0$. A loss $g(a) < 0$ serves as an entrance fee. Then the requirement $x \geq g(a)$ serves as a lower bound. However, this is pre-conditioned by a positive flow $x > 0$.

Many flow problems in standard and in generalize networks can be formulated by linear programs, where a real-valued variable for each arc expresses the amount of flow on the arc. Additive losses and gains need additional 0–1 variables to express the use of an arc.

By the losses and gains of the arcs the amount of flow changes from the source towards the sink, and the change depends on the used arcs. Thus the maximum flow can be measured at the source or at the sink reflecting the producer's or the consumer's view. This makes an essential difference for the amount of flow and also for the maximum flow problem. We define the out-flow and in-flow problems, which are reversible if losses and gains are interchanged and do not consume all flow.

In many respects networks with additive losses and gains are different from standard and generalized networks. First, flows are not scalable on a path with a free capacity. If an arc a has not yet been used, the new flow along a increases by the gain $g(a)$, or it decreases and must exceed at least $-g(a)$ if the arc is lossy. This is a discontinuity property. Another drawback of additive networks is the lack of *duality* with the classical max-flow, min-cut theorem and flow augmentations [1,7,10]. Due to the losses and gains the maximum flow in a network is no more static and the size of minimal cuts may vary. There are networks and flows with augmenting s–t paths, but the paths cannot be used to augment the flow, since the gain of an unused arc may lead to an excess at its end node or a loss may exceed the available flow.

The shortest path problem in additive networks is the min-cost flow problem of a unit flow from the source on a single path in uncapacitated networks. The losses and gains change the amount of flow such that the cost of an arc on the path is the product of the cost per unit and the amount of flow at its entrance. The flow may dry out on the path, and there are lossy cycles, which give raise to the path and the flow models.

Moreover, the shortest path property is no more hereditary. It is no more true that the sub-path, the prefix or the suffix of a shortest path is a shortest path. These properties are indicators for the intractability of networks with additive losses and gains.

This paper is organized as follows. In Section 2 we introduce networks with losses and gains, and in Section 3 we investigate the shortest path problem. In networks with losses and gains there are two versions: a flow model and a path model. We study some basic properties of shortest paths in the path model and show that the shortest path problem is $\mathcal{NP}$-hard in both models, and is solvable in pseudo-polynomial time for nonnegative costs and gains by a dynamic programming approach.

In Section 4 we address the maximum flow problem in additive networks and show that it is $\mathcal{NP}$-hard in the strong sense by a reduction from Exact Cover by 3-Sets [11]. These $\mathcal{NP}$-hardness results hold for the out-flow from the source, even for networks with integral capacities and with unit gain or with loss two for each arc, and for the in-flow into the sink, even for networks with unit loss or with unit gain for each arc. Moreover, the maximum flow problem is MAX-SNP-hard, and

is hard to approximate. On the contrary, in unit-loss networks the maximum flow problem from the source can be solved efficiently by the Edmonds–Karp algorithm in $\mathcal{O}(nm^2)$.

In summary, our results reveal an essential difference between networks with additive and with (or without) multiplicative losses and gains. From the algorithmic point additive networks are much harder. Here the common flow problems are intractable whereas they are tractable in generalized networks with multiplicative losses and gains and in standard networks.

2. Networks with losses and gains

In this section we introduce the main concepts on networks with additive losses and gains and establish basic properties of flows and paths.

A *network with losses and gains* is a tuple $N = (V, A, s, t, c, u, g)$, where (V, A) is a directed graph with n nodes and m arcs and two distinguished nodes, a source s and a sink t without incoming and outgoing arcs, respectively. Each arc a has parameters for the cost, the capacity, and the gain, respectively, which are defined by weighting functions. $c(a)$ is the *cost* per unit flow, and the *capacity* $u(a)$ is an upper bound on the amount of flow with $u(a) \geq 0$. Both are taken at the entrance of the arcs. g assigns a gain function g_a to each arc a, where g_a is a function over the nonnegative reals with $g_a(x) \geq 0$ for $x \geq 0$ and $g_a(0) = 0$. The *gain* g_a may vary the amount of flow if there is a flow. If $x > 0$ units of flow enter a, then $g_a(x)$ units exit provided that $x \leq u(a)$. An arc is *gainy* if $g_a(x) > x$, *conserving* if $g_a(x) = x$, and *lossy* if $g_a(x) < x$. These terms extend to paths and cycles. For convenience, conserving arcs are lossy or gainy. The gain of an arc a is *additive* if $g_a(x) = x + g(a)$, and *multiplicative* if $g_a(x) = g(a) \cdot x$ for some real (or integer) $g(a)$. More general gain functions are not discussed here.

For convenience we speak of additive and multiplicative arcs and networks, and we shall restrict ourselves to these cases. Moreover, we specialize to additive networks with integral capacities and integral losses and gains. Such a network is a *unit-loss network* if the arcs have no or unit loss, and a *unit-gain network* if the gains are $g(a) \in \{0, 1\}$. For an integer $k \geq 1$ it is a *strictly k-loss network* if each arc has loss k and a *strictly k-gain network* if each arc has gain k. For $k = 1$ these are *strictly unit-loss* and *strictly unit-gain* networks.

A *flow* in a network N is a nonnegative real-valued function $f : A \to \mathbb{R}^+$ on the arcs satisfying the capacity constraints $0 \leq f(a) \leq u(a)$ and flow conservation at each node except at the source and the sink, and which respects the gains of the arcs. If $f(a) = 0$ the arc a is unused. Flow conservation at a node v is expressed by the equation $\sum_{a=(v',v)} g_a(f(a)) = \sum_{a'=(v,v'')} f(a')$, where all terms $g_a(f(a))$ and $f(a')$ are nonnegative.

The *out-flow* is the amount of flow from the source $f_{out} = \sum_{a=(s,v)} f(a)$, and the *in-flow* is the flow into the sink $f_{in} = \sum_{a=(w,t)} g(f(a))$. These values may differ in networks with losses and gains. The cost of a flow is the sum of the cost of the arcs, where the cost of each arc is charged per unit of flow, $c(f) = \sum_a c(a)f(a)$.

If flow x with $x > 0$ arrives at an unused arc a, then the flow must satisfy the capacity constraint $x \leq u(a)$ at the entrance. If in addition the arc is lossy with $g(a) < 0$ the flow must compensate the loss, i.e., $x \geq -g(a)$. If $x = -g(a)$, all flow is consumed by the arc. If an arc is unused the loss has no effect. A lossy arc cannot be modeled by an arc with a lower bound branching off the original arc, since the loss is conditioned by a positive flow whereas a lower bound is permanent. This distinguishes losses from lower bounds.

The *maximum out-flow problem* maximizes f_{out} and the *maximum in-flow problem* maximizes f_{in}. In standard networks these values coincide and in generalized networks with multiplicative losses and gains there are polynomial transformations between these problems [12,22]. However, in additive networks (even with unit losses and gains) these problems are essentially different, as shall be shown in Section 4.

The shortest path problem is another major problem. Shortest paths have many applications of their own and are often used as a subproblem, e.g. for the maximum flow and the min-cost flow problem.

In networks with losses and gains shortest paths must be taken with care. There are two models: a path model and a flow model. In the path model, an *s–t path* consists of a finite sequence of arcs $a_i = (v_{i-1}, v_i)$ for $i = 1, \ldots, k$ with $s = v_0$ and $t = v_k$. It is *simple*, if all nodes are distinct. The cost of a path τ is the cost of a unit flow from the first node and is the sum of the accumulated cost of the arcs of τ, provided that the path is feasible, i.e., the flow on each arc is nonnegative. If all incoming flow is consumed by an arc, then the subsequent arcs are free of charge. The feasibility condition may lead to complex shortest paths consisting of simple paths between gainy and lossy cycles. In particular, if a path includes a cycle then the losses and gains of the arcs in the cycle are accumulated repeatedly for each use of an arc.

Alternatively, in the flow model the cost of a path is the cost of a unit out-flow in an uncapacitated network. Here the gain function of each arc is used at most once, even in a cycle. Such paths play an important role in generalized networks with multiplicative losses and gains, and consist of a simple s–t path or of a simple path and are followed by a lossy cycle where all flow is consumed [4,12,25].

We focus on the complexity and show that the shortest path problem is $\mathcal{NP}$-hard in either model of additive networks, even in directed acyclic networks, where both models coincide. Moreover, we provide a pseudo-polynomial time solution with nonnegative costs and gains.

Flows and paths in additive networks are illustrated by an example. The capacity and the gains are annotated below the arcs. The numbers above the arcs are the flow values at the entrance and at the exit. All arcs have unit cost $c(a) = 1$. The unit flow from s gains one unit and due to the capacities splits into an upper flow through the nodes 1, 2, 3, t and a median

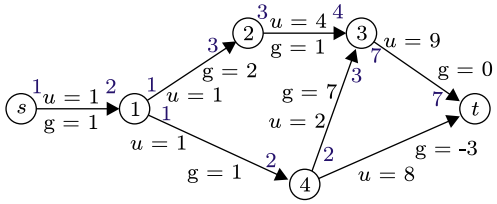

Fig. 1. Paths and flows in an additive network.

flow through 1, 4, 3, t. A lower path through the nodes 1, 4, t is infeasible, since the arc $(4, t)$ has a loss of three units and needs at least three units of flow at its entrance. The upper path from 1 to t delivers 4 units of flow at t and has cost 8. If the median path comes next it adds 3 units at the cost of 6 (Fig. 1).

3. Shortest paths

Here we investigate some basic properties of shortest (or cheapest) paths in additive networks and establish the $\mathcal{NP}$-hardness of the shortest path problem, both in the path and in the flow model. The path model is considered first.

Definition 1. Let N be an additive network with cost $c(a)$ and gain $g(a)$ for each arc a. Let $\tau = (a_1, \ldots, a_r)$ be a path with the arcs $a_1, \ldots, a_r$. For an initial seed $g_0 > 0$ and $0 \le j \le r$ let $\gamma(\tau, j, g_0) = \sum_{i=1}^{j} g(a_i) + g_0$ be the prefix sum of the gains with seed g_0.

A path τ is full if $\gamma(\tau, j, 1) > 0$ for $j = 1, \ldots, r$. τ is a dead end if there is some $q \le r$ with $\gamma(\tau, j, 1) > 0$ for $j = 1, \ldots, q-1$ and $\gamma(\tau, q, 1) = 0$. τ is infeasible if there is some $q \le r$ with $\gamma(\tau, q, 1) < 0$ and $\gamma(\tau, j, 1) > 0$ for $j = 1, \ldots, q-1$.

A full path has a positive flow at each arc with a unit as the initial seed. In a dead end there is an arc which absorbs all flow arriving at its entrance. Such paths are *feasible*. A path is infeasible if the flow were negative.

Definition 2. The additive gain of a full path $\tau = (a_1, \ldots, a_r)$ is $g^+(\tau) = \gamma(\tau, r, 1)$ and is zero for a dead end. It is undefined for infeasible paths.

The additive cost of a full path τ is $c^+(\tau) = \sum_{i=1}^{r} c(a_i)\gamma(\tau, i-1, 1)$. If τ is a dead end with $q = min\{j \mid \gamma(\tau, j, 1) = 0\}$, then $c^+(\tau) = \sum_{i=1}^{q} c(a_i)\gamma(\tau, i-1, 1)$. A path between two nodes with minimum cost is called a *shortest path*. The usual *length* of a path τ is the cost without additive gains $l(\tau) = \sum_{i=1}^{r} c(a_i)$.

Thus, the additive gain of a feasible path is the sum of gains of the arcs and is set to zero if the path is a dead end. The cost is the accumulated cost of the arcs, which are charged per unit at the entrance.

The feasibility of a path $\tau = (a_1, \ldots, a_r)$ may depend on the initial seed, which is the flow at the entrance of the first arc a_1. There is a threshold $T = min\{\sum_{i=1}^{j} g(a_i) | 1 \le j \le r\}$ such that τ is full if the initial seed exceeds $-T$; τ is a dead end at $-T$ and is infeasible below. A higher initial seed increases the cost of a path linearly with its length.

Some basic properties of paths in additive networks can be retrieved from paths with a common prefix or a common suffix, which follow directly from the given definitions.

Lemma 1. *If a full path $\tau = \tau_1 \circ \tau_2$ is the composition of two full sub-paths the gain is $g^+(\tau) = g^+(\tau_1) + g^+(\tau_2)$ and the cost accumulates to $c^+(\tau) = c^+(\tau_1) + c^+(\tau_2) + g^+(\tau_1)l(\tau_2)$.*

If $\tau = \tau_1 \circ \tau_2$ is a full path, then the prefix τ_1 is a full path. However, τ_2 may be full, a dead end, or infeasible. If τ is a dead end, then τ_1 may be full or a dead end, and τ_2 may be full, a dead end or infeasible.

For a comparison of the cost of paths with a common prefix or a common suffix we use two parameters, the cost and the gain, or the cost and the length. This is used in our pseudo-polynomial time algorithm.

Lemma 2. *Let $\tau_1 = \rho \circ \sigma_1$ and $\tau_2 = \rho \circ \sigma_2$ be full paths with the same prefix and full suffixes. Then*
$g^+(\tau_1) - g^+(\tau_2) = g^+(\sigma_1) - g^+(\sigma_2)$, *and*
$c^+(\tau_1) - c^+(\tau_2) = c^+(\sigma_1) - c^+(\sigma_2) + g^+(\rho)(l(\sigma_1) - l(\sigma_2))$.

Lemma 3. *Let $\tau_1 = \sigma_1 \circ \rho$ and $\tau_2 = \sigma_2 \circ \rho$ be full paths with the same suffix.*
$g^+(\tau_1) - g^+(\tau_2) = g^+(\sigma_1) - g^+(\sigma_2)$, *and*
$c^+(\tau_1) - c^+(\tau_2) = c^+(\sigma_1) - c^+(\sigma_2) + l(\rho)(g^+(\sigma_1) - g^+(\sigma_2))$.

In standard networks each sub-path of a shortest path is itself a shortest path. Thus shortest is hereditary. This property is important for the efficient computation of shortest paths by dynamic programming techniques. It does not hold with losses and gains. Since the multiplicative gain of a finite path is positive, we can conclude that each suffix of a shortest path is a shortest path. A similar property does not hold for prefixes. The algebraic structure is a distributive closed semi-ring, as discovered by Oldham [25] and independently by Batagelj [3]. In additive networks the situation is even worse. The suffix

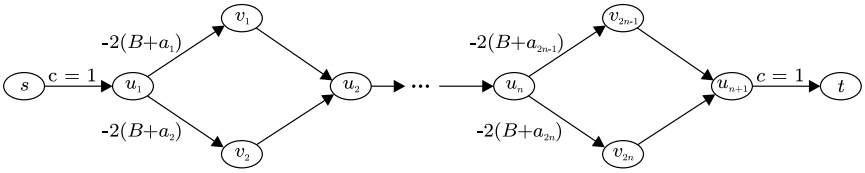

Fig. 2. Reduction from PARTITION.

of a path is not necessarily feasible, since the gains may run into the negative, and the prefixes or the suffixes of a shortest path are not necessarily shortest paths. This can be shown by simple examples.

Now we turn to shortest paths in additive networks. For the reachability of a node v it must be checked that v is reachable by a full path from the source. This can be done by a Bellman–Ford-type algorithm using the distance function $d(a) = -g(a)$ and the initial distance $d(s) = -1$ at the source such that the intermediate distances of the nodes are negative. If this algorithm uses a direct negative cycle detection strategy [5,23,26] and detects a cycle of negative cost at some node v, then all graph theoretic successors of v are reachable in the path model, where the gain is accumulated at every visit of an arc. In the flow model the used arcs are marked and the gains are taken once.

The reachability can be computed in $\mathcal{O}(nm)$, whereas the shortest path problem is practically unfeasible.

Theorem 1. *The shortest path problem in additive networks with losses and gains is $\mathcal{NP}$-hard.*

Proof. We reduce from the PARTITION problem [11]. Given a finite ordered set of positive integers $A = \{a_1, a_2, \ldots, a_{2n}\}$, is there a subset $A' \subseteq A$ such that A' contains exactly one of a_{2i-1}, a_{2i} for $1 \le i \le n$ and $\sum_{a_i \in A'} a_i = \sum_{a_i \notin A'} a_i$?

Let $2B = \sum_{a_i \in A} a_i$ and assume B to be an integer. We construct an instance of the shortest path problem in additive networks as follows: Let $N = (V, E, s, t, c, u, g)$ be an additive network as shown in Fig. 2. $V = \{s, t\} \cup \{u_1, \ldots, u_{n+1}\} \cup \{v_1, \ldots, v_{2n}\}$, $E = \{(s, u_1), (u_{n+1}, t)\} \bigcup_{i=1}^{n} \{(u_i, v_{2i-1}), (u_i, v_{2i}), (v_{2i-1}, u_{i+1}), (v_{2i}, u_{i+1})\}$.

Define the gain of the arcs by $g((s, u_1)) = 2B(n+1)$, $g((u_i, v_j)) = -2(B + a_j)$ for $j = 2i-1, 2i$ and $i = 1, \ldots, n$, and $g(a) = 0$ otherwise, let the capacity be $1 + 2B(n+1)$ for every arc and let the cost be $c((s, u_1)) = c((u_{n+1}, t)) = 1$ and $c(a) = 0$ otherwise.

Now we claim that N has a shortest s–t path with cost 2 if and only if the instance of PARTITION has a solution. First suppose that PARTITION has a subset A' as required, then there is a path π from s to t through v_i if and only if $a_i \in A'$. The gain $2B(n+1)$ from the first arc is equal to the total loss $\sum_{a_i \in A'} 2(B + a_i)$, and the flow on (s, u_1) and on (u_{n+1}, t) is equal to 1 so that $c^+(\pi) = 2$.

Conversely, if N has a shortest s–t path π with cost 2, then PARTITION has the required subset A'. Note that N is acyclic and that all losses and gains are even. If one unit of flow leaves s, then at least one unit must reach t. Conversely, if a unit flow arrives at t, then at least one unit leaves s. Hence, the path cannot be a dead end, and the cost of π is at least 2. Since π has cost 2, one unit leaves s and one unit reaches t, which implies that the total loss of π equals the gain $2B(n+1)$ of the first arc. Clearly, π passes exactly one of v_{2i-1}, v_{2i} for $i = 1, \ldots, n$. Accumulate these nodes into the set $V(\pi)$ and set $A' = \{a_i \in A \mid v_i \in V(\pi)\}$. Then $\sum_{a_i \in A'} a_i = B$ derives from $\sum_{v_i \in V(\pi)} 2(B + a_i) = 2(n+1)B$, and A' is the desired subset.

The construction works equally well in the path and in the flow model, since the network is acyclic. □

The used PARTITION problem is not $\mathcal{NP}$-hard in the strong sense [11]. The restriction to simple paths is harder by the reduction from the direct Hamilton path problem.

Theorem 2. *The simple shortest path problem in additive networks is $\mathcal{NP}$-hard in the strong sense. It remains NP-hard even if all numbers are integers with absolute values bounded by $\mathcal{O}(n^2)$ and either the costs or the gains are nonnegative.*

Proof. We reduce from the directed Hamilton path problem [11]. Let G be an instance of the Hamilton path problem with source s and sink t. Construct a network N from G by node splitting. For each node v of G there are two nodes v_1 and v_2 with the internal arc (v_1, v_2). v_1 inherits all incoming arcs and v_2 all outgoing arcs, such that (u, v) is an arc in G if and only if (u_2, v_1) is an arc in N. These arcs have zero cost and zero gain. Finally, add an arc (t_2, t^*) towards a new sink t^*.

First, with nonnegative gains the internal arcs (v_1, v_2) have unit cost and gain one, and the arc (t_2, t^*) has cost $-n$, where n is the size of G. This is the only negative cost arc. A simple path from s to t through k nodes in G has an associated path in N with gain k and cost $k(k-1)/2$ from s_1 to t_2. The total cost is $k(k-1)/2 - kn$, which is minimum for $k = n$, which holds if and only if the simple path is a Hamilton path.

With nonnegative cost add a new source s^* and the arc (s^*, s_1) with zero cost and gain $n+1$. Each inner arc (v_1, v_2) has unit cost and gain -1, and the arc (t_2, t^*) has cost $(n+2)(n+1)$. This arc contributes $(n+2)(n+1)$ to the cost of the path if and only if the gain has decreased to one, which holds, if and only if there is a Hamilton path in G.

Clearly, by guess and check it can be determined whether or not there is a path or a simple path with cost at most K from the source to some node v. □

These $\mathcal{NP}$-hardness results add to the *NP*-hardness of the cheapest simple path problem in multiplicative networks [4]. Both are $\mathcal{NP}$-hard in the strong sense.

The $\mathcal{NP}$-hardness results rely on losses and gains. With nonnegative costs and gains shortest paths are simple and full and can thus be considered likewise in the path or in the flow model. There is a pseudo-polynomial time solution by a dynamic programming approach, which leads to a polynomial time solution of the shortest path problem if the gains or the costs are polynomially bounded in the size of the network.

Theorem 3. *With nonnegative costs and gains the shortest path problem in additive networks can be solved in pseudo-polynomial time.*

Proof. Since the gains are nonnegative all paths are full, and shortest paths are simple, since a cycle increases the cost.

By Lemma 1 the cost of a simple path $\tau = \tau_1 \circ \tau_2$ is $c(\tau) = c(\tau_1) + c(\tau_2) + g(\tau_1)l(\tau_2)$, where $l(\tau_2)$ is the length of τ_2 (or the cost without gains). Since the cost and the gain of the arcs are nonnegative, there is a cheaper path $\tau_1' \circ \tau_2$, if τ_1' is cheaper than τ_1 in both the parameters, the cost and the gain.

A pair of numbers (x, y) *dominates* another pair (x', y') if $x \geq x'$ and $y \geq y'$. A *minimal pair* does not dominate any other pair. If the pair (x, y) represents the cost and the gain of a path from the source to some node v, or the cost and the length of a path from v to the sink, and the costs and the gains are nonnegative, then a dominating pair induces higher costs. Thus only minimal pairs are needed. First, consider paths from the source. If $0 \leq g(a) \leq p(n)$ and the gains are integers then there are at most $(n-1)p(n)$ values for the gains of the nodes on simple paths from the source. For each such value y compute the shortest path from the source with gain at most y by a standard algorithm. At each node there are at most $(n-1)p(n)$ minimal pairs (x, y) with cost x and gain y. For each gain y the cost of the shortest path from the source s to all nodes v can be computed by a dynamic programming approach using the equations $c_0(s) = 0$ and $c_y(v) = \min\{c_{y-g((u,v))}(u)+c((u, v))(y-g((u, v)))|$ there is an arc $a = (u, v)$ with cost $c((u, v))$ and gain $g((u, v))\}$, where $c_y(v)$ is the least cost of a path from s to v with gain y, and finally taking the minimum over all y.

Accordingly, for integral costs in the range $0 \leq c(a) \leq p(n)$, the length of a shortest path is bounded by $(n-1)p(n)$. Now consider pairs of the cost and the length of paths from nodes to the sink. The dynamic programming approach is $c_0(t) = 0$ and $c_y(v) = min\{c(a) + g(a)c_{y-c(a)}(w)|$ there is an arc $a = (v, w)$ with cost $c(a)$ and gain $g(a)\}$, where $c_y(v)$ is the least cost of a path from v to t with length exactly y.

The running time of the algorithms is polynomial in the size of the graph and $p(n)$, i.e. pseudo-polynomial. □

Theorem 4. *The shortest path problem in additive networks can be solved in polynomial time if the costs and the gains are nonnegative and either the costs or the gains are nonnegative integers and bounded from above by some polynomial $p(n)$ in the size of the graph.*

Note that the shortest path problem in additive networks is computationally equivalent to the shortest weight-constrained path problem [11].

4. Maximum flow problems in additive networks

In networks with losses and gains the maximum flow problem can be stated in various ways. There are networks with a distinguished source without incoming arcs and a distinguished sink without outgoing arcs, and circulations with flow conservation at all nodes except the source (or the sink). Due to the gains the out-flow from the source and the in-flow into the sink are generally different. The maximum flow can be measured at the source or at the sink (or at any intermediate minimum cut). Both quantities can be combined into the *profit* problem, which aims at maximizing the difference of the q-fold in-flow and the out-flow. In standard and in multiplicative networks there are linear time reductions between these versions [12,22]. In additive networks the situation is different. Although all general versions of maximum flow problems are shown to be $\mathcal{NP}$-hard, there are no linear time reductions, unless $\mathcal{P} = \mathcal{NP}$. This is due to the fact that the maximum out-flow problem is solvable in polynomial time with unit-loss arcs and is $\mathcal{NP}$-hard with unit-gain arcs, and the maximum in-flow problem is $\mathcal{NP}$-hard even with unit losses or with unit gains.

Losses and gains are defined on the arcs. This is no real restriction of the general case with losses and gains for nodes and arcs. Nodes and arcs can model losses and gains mutually by splitting vertices and by replacing arcs by paths of length 2.

The common network flow problems have an LP-formulation and can be solved by general purpose LP-solvers and by the Simplex algorithm [1]. For each arc $a = (i, j)$ define a variable x_{ij} such that x_{ij} represents the amount of flow on the arc (i, j). Then $0 \leq x_{ij} \leq u_{ij}$ express nonnegative flow and the capacity constraints. There are equations for the nodes expressing flow conservation. The resulting node–arc incidence matrix has entries -1 if an arc is leaving node i, $+1$ if an arc is entering node j, and 0, otherwise. In generalized networks with multiplicative gains the -1s are replaced by $-g_{ij}$, where g_{ij} is the gain factor of the arc (i, j). Observe that this formulation does not capture lossy and gainy cycles which consume or generate all flow.

For a linear programming formulation of the maximum flow problem in an additive network there is a real-valued variable x_{ij} for the amount of flow on the arc (i, j) and a 0–1 variable y_{ij} which expresses the use of the arc (i, j) by some positive flow.

For the source s maximize $\sum x_{sj}$ subject to

(i) $\sum_i x_{ij} + \sum_i g_{ij}y_{ij} = \sum_k x_{jk}$ for each j with $j \neq s, t$
(ii) $0 \leq x_{ij} \leq u_{ij}$

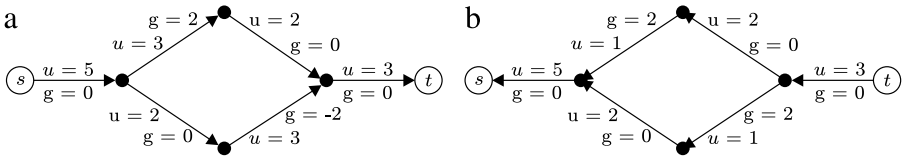

Fig. 3. A non-reversible network.

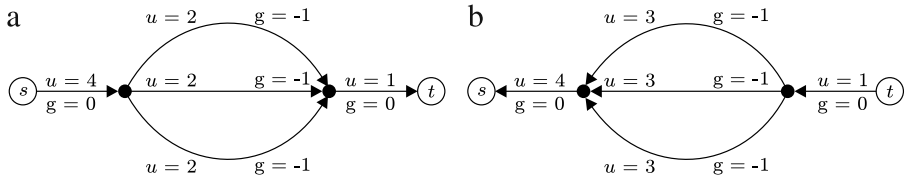

Fig. 4. Network with fractional flows.

(iii) $x_{ij}/u_{ij} \le y_{ij} \le x_{ij}K$ for some sufficiently large positive number $K = U/\epsilon$, where $U = \sum_{i,j} u_{ij}$ and $\epsilon > 0$ is the value of the least significant digit of the capacities u_{ij} and gains g_{ij}, and

(iv) $x_{ij} \ge -g_{ij}y_{ij}$.

(i) expresses flow conservation at the node j other than the source and the sink. (ii) are the capacity constraints of the flow at the entrance of the arcs. (iii) is the novelty for additive networks and expresses "no flow, no gain" by $x_{ij} > 0 \Leftrightarrow y_{ij} > 0$, and (iv) expresses the feasibility of a flow at a lossy arc. If $g_{ij} < 0$, then the flow x_{ij} at the arc must exceed the loss.

This ILP can be solved by the usual relaxation for the 0–1 variables to $0 \le y_{ij} \le 1$ and branch-and-cut techniques.

Observe the difference between additive losses and gains and lower bounds for a flow. Lower bounds on the arcs are static and must always be satisfied by the flow, whereas losses and gains are dynamic and are activated if there is a positive flow.

In additive networks there is no symmetry between the generation and the consumption of flow and the in-flow and out-flow problems. All flow is generated in the single source, whereas flow is consumed in the sink and in lossy arcs, if the loss equals the incoming flow. These roles are interchanged in the reverse network.

Definition 3. For an additive network N construct the *reverse network* N' by reversing all arcs and exchanging the source and the sink. The capacity and the gains of the arcs are adjusted. If $a = (v, w)$ is an arc of N with capacity u and gain g, then the reverse arc $a' = (w, v)$ has capacity $u + g$ and gain $-g$. Thus N' is lossy (gainy) if N is gainy (lossy). Cost functions are discarded here.

Lemma 4. *Let f be a flow in an additive network N such that for each arc a, $f(a) + g(a) > 0$ if $f(a) > 0$. Then there is a flow f' in the reverse network N' with $f_{out} = f'_{in}$ (and $f_{in} = f'_{out}$).*

Proof. By the assumption the flow f is either positive or zero at both ends of the arcs. Thus all flow is consumed in the sink and the flow can be reversed in N'. If $f(a)$ is the flow on the arc a, then $0 < f(a) \le u(a)$ and $f(a) + g(a) > 0$ by the assumption. In the reverse network N' a flow of size $f(a) + g(a)$ enters the reverse arc and $f(a)$ units exit. Hence $f_{out} = f'_{in}$ and $f_{in} = f'_{out}$. □

For the exchange of the direction in Lemma 4 we require a positive flow at both ends of the used arcs. This restriction cannot be dropped, as the network in Fig. 3 illustrates. In N there is a maximum out-flow of five units, where the flow on the lower path dries out and one unit arrives at t. In the reverse network there is flow only along the upper path and three units arrive at s.

Moreover integrality is no longer preserved, even with integral parameters. The network N from Fig. 4 may send two units of flow along the upper path and one unit along the middle and the lower paths. In the reverse network N' there is a maximal in-flow at s, if the unit flow splits and sends positive fractions along each path.

We now establish our $\mathcal{NP}$-hardness results for the maximum flow problems.

Theorem 5. *In additive networks the maximum out-flow problem is $\mathcal{NP}$-hard in the strong sense, even in strictly unit-gain networks. Accordingly, the maximum in-flow problem in strictly unit-loss networks is $\mathcal{NP}$-hard in the strong sense.*

Proof. We reduce from the Exact Cover by 3-Sets problem, X3C [11]. Let (X, B) be an instance of X3C with $|X| = 3q$. The network N has two extra nodes s and t. For each 3-set $b = \{x, y, z\}$ there is a node b in N and arcs (b, x), (b, y), (b, z) with unit capacity and an arc (s, b) from the source s with capacity 2. Finally, for each element $x \in X$ there is a node x in N and an arc (x, t) with capacity 2. All arcs have unit gain.

Fig. 5 illustrates the network and the reduction.

We claim that the maximum out-flow is $2q$ if and only if there is a solution of X3C. First, if there is a solution of X3C, then there is the out-flow of size $2q$ with 2 units from s to each node b of the 3-sets in the solution, and this arc (s, b) gains one

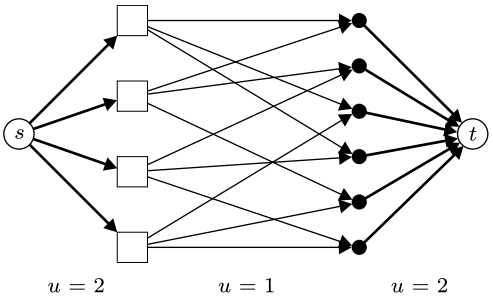

Fig. 5. A strictly unit-gain network for out-flow and X3C.

unit such that 3 units arrive at b. These split into three unit flows through each of the nodes x, y, z of the elements of b and grow by two units towards t. Hence $9q$ units arrive at t.

Conversely, suppose there is a flow with $f_{out} \geq 2q$ and k arcs from s with a positive flow, then $f_{out} \leq 2k$ and $k + f_{out} \leq 3q$ by the capacity constraints, implying that $k = q$ and $f_{out} = 2q$. Let $(s, b_{i_1}), \ldots, (s, b_{i_q})$ be the arcs from s with a positive flow. Then the arcs incident with $b_{i_1}, \ldots, b_{i_k}, t$ are saturated and $b_{i_1}, \ldots, b_{i_k}$ define a solution of the X3C instance.

For the maximum in-flow problem in unit-loss networks the reverse network is used. □

There is an asymmetry between the maximum out-flow and in-flow problems in non-reversible networks. The maximum in-flow problem remains $\mathcal{NP}$-hard even with unit gains and the maximum out-flow problem remains $\mathcal{NP}$-hard even with loss two for all arcs. However, it is solvable in polynomial time in unit-loss networks with loss zero or one, as shown in Theorem 11.

Theorem 6. *In additive networks the maximum out-flow problem is $\mathcal{NP}$-hard in the strong sense, even in strictly 2-loss networks.*

Proof. In a similar way as above we reduce from the Exact Cover by 3-Sets problem. Let (X,B) be an instance of X3C with $|X| = 3q$.

For every element x there is a node x and an arc (s, x) with capacity 5. For every 3-set $b = \{x, y, z\}$ there are two nodes b and its successor b' and arcs (x, b), (y, b), (z, b), (b, b'), (s, b') all with capacity 3, and an arc (b', t) with capacity 2, respectively. All arcs have loss 2; see Fig. 6.

There is a solution of X3C if and only if there is an out-flow of size $16q + 2|B|$. Given the solution of X3C, the arcs towards the chosen 3-sets are saturated. 5 units are sent to the node of each element, where 3 units arrive. These are sent to its 3-set, where one unit arrives. So $15q$ units leave the source on these arcs. From each chosen 3-set 3 collected units are sent to its successor b', and each such b' receives one more unit directly from the source, and $3q$ units leave the source on these arcs. There is a flow of 2 units on each of the remaining $|B| - q$ arcs from s to a successor b', which is immediately consumed. Thus $f_{out} = 15q + 3q + 2(|B| - q)$, and all flow is consumed towards t.

Conversely, suppose there is a flow with $f_{out} \geq 16q + 2|B|$. Since the capacity of the arcs from the source to nodes of elements is $15q$, there are $k \geq q$ arcs $(s, b'_{i_1}), \ldots, (s, b'_{i_k})$ from s to the successors of 3-sets with a flow greater than two. Let $C' = \{b'_{i_1} \ldots, b'_{i_k}\}$ be the set of these nodes and let C be the predecessors. Each node $b' \in C'$ receives a positive flow. Since the arcs from the successors b' to the sink consume two units of flow and the arcs (b, b') have capacity 3 and loss 2, the arcs incident with $b' \in C'$ are saturated. Hence, each node $b \in C$ receives 3 units of flow, and $15k$ units of flow are sent from s to these nodes. It follows that $15k \leq 15q$ and thus $k = q$ and $18k$ units of flow are sent from s to the nodes $b' \in C'$. The remaining flow of $2(|B| - q)$ units is directly sent to the successor nodes with $b' \notin C'$, and each such arc consumes two units. Now the 3-sets with $b \in C$ form the solution of X3C. □

Theorem 7. *In additive networks the maximum in-flow problem is $\mathcal{NP}$-hard in the strong sense, even in strictly unit-gain networks.*

Proof. Again we reduce from X3C with an instance (X, B) and $|X| = 3q$. All arcs are unit-gain arcs.

For every element x there is a node x and an arc (x, t) to the sink t with capacity two. For every 3-set $b = (x, y, z)$ there are two nodes b, b' and arcs (s, b) and (b, b') with capacity 1 and 2, respectively. Between the 3-sets and the elements there are arcs (b', x), (b', y), (b', z) with unit capacity; see Fig. 7.

There is a solution of X3C if and only if the in-flow is $9q$. Given a solution of X3C, a unit flow is sent to the first node of each chosen 3-set and is increased such that three units arrive at b'. From there a unit is sent to each element and is increased by one unit such that two units arrive at each of the $3q$ elements. One more unit is added towards the sink such that the maximum in-flow of $9q$ units arrives at t, and all arcs to the sink are saturated.

Conversely, suppose that $9q$ units arrive at t. Then all arcs from the elements x to the sink are saturated. Each x receives a flow from at most one of its predecessors, since each such arc is a unit-gain arc and two or more predecessors give a flow of more than two units. For the maximum in-flow each x receives one unit from a single predecessor b'. At each such b' there is a flow f with $2 < f \leq 3$. f must be integral for a maximum in-flow and thus $f = 3$. Hence, the b's with unit flow from the source form a solution of X3C. □

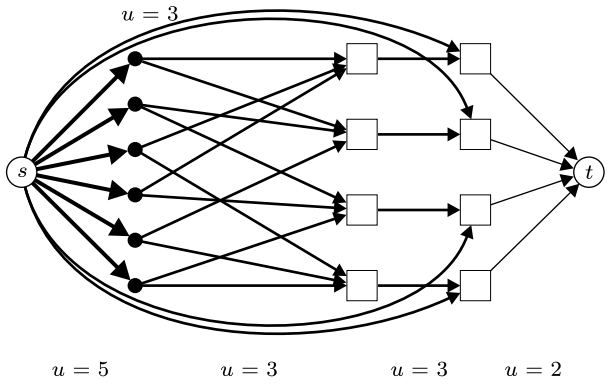

Fig. 6. An additive network with loss 2 for out-flow and X3C.

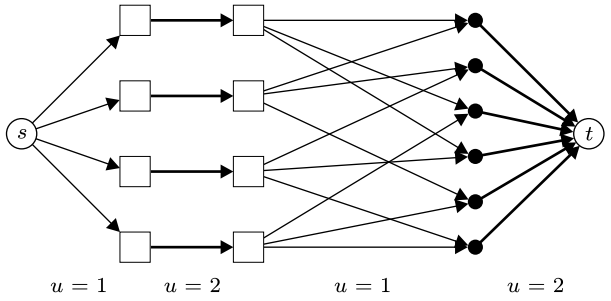

Fig. 7. A unit-gain additive network for in-flow and X3C.

The maximum flow problems are $\mathcal{NP}$-hard, and they are MAX-SNP-hard. This is a direct consequence of the above reductions. As a consequence, maximum flow is hard to approximate to within $(1-\epsilon)$ and there is no polynomial time approximation scheme. This holds for all restricted versions from the previous theorems.

Theorem 8. *In additive networks the maximum flow problem from the source is MAX-SNP-hard, even in strictly unit-gain networks or in strictly 2-loss networks. Accordingly, the maximum flow problem into the sink is MAX-SNP-hard, even in strictly unit-loss or in strictly unit-gain networks.*

Proof. The optimization version of Exact Cover by 3-Sets is MAX-SNP-hard [20]. Consider the reductions from the proof of Theorems 5–7. These are L-reductions, such that $3q$ elements are covered if and only if there is flow of size $2q$, $16q + 2|B|$, and $9q$, respectively. □

Moreover, we have a further in-approximation result.

Theorem 9. *There exists a fixed $\epsilon > 0$ such that approximating the maximum out-flow problem within a factor of n^ϵ in additive networks is $\mathcal{NP}$-hard, where n is an upper bound on the maximum out-flow.*

Proof. Approximating the MAX-SET-PACKING problem within a factor of n^ϵ is $\mathcal{NP}$-hard for some fixed $\epsilon > 0$, where n is the cardinality of the basic set [2,16,30]. For the proof it suffices to describe an L-reduction τ from the MAX-SET-PACKING problem to the maximum out-flow problem in additive networks.

Consider an instance $\mathcal{I}$ of MAX-SET-PACKING: Given a collection $\mathcal{C} = \{C_1, \ldots, C_m\}$ of subsets of $S = \{e_1, e_2, \ldots, e_n\}$, find a subcollection of disjoint subsets $\mathcal{C}' \subseteq \mathcal{C}$ such that $|\mathcal{C}'|$ is maximized.

First we construct an instance $\tau(I)$ of the maximum out-flow problem in an additive network $N = (V, E, u, g)$ as follows.

$$V = \{s, t, u_1, \ldots, u_m, v_1, \ldots, v_n\},$$

$$E = \{(s, u_1), \ldots, (s, u_m)\} \cup \{(v_1, t), \ldots, (v_n, t)\} \bigcup_{i=1}^{m} \{(u_i, v_j) | e_j \in C_i\},$$

$$u(a) = 1 \quad \text{for every arc } a,$$

$$g(a) = \begin{cases} |C_i| - 1 & \text{if } a = (s, u_i) \text{ and } 1 \le i \le m, \\ 0 & \text{otherwise.} \end{cases}$$

This completes the construction of $\tau(I)$. For a feasible solution f of $\tau(I)$, let $\mu(\tau(I), f)$ denote the objective function value of f, and let $\mu(I)$ and $\mu(\tau(I))$ denote the optimal values of problems I and $\tau(I)$, respectively. We claim that

$$\mu(I) = \mu(\tau(I)).$$

First, for a maximum set packing $\mathcal{C}^* \subseteq \mathcal{C}$, we define a solution $\bar{f} = (\bar{f}(a))$ of $\tau(I)$ as follows.

$$\bar{f}(a) = \begin{cases} 1 & \text{if } a = (s, u_i) \text{ and } C_i \in \mathcal{C}^*, \\ 1 & \text{if } a = (u_i, v_j) \text{ and } e_j \in C_i \in \mathcal{C}^*, \\ 1 & \text{if } e = (v_j, t) \text{ and } e_j \in C_i \in \mathcal{C}^*, \\ 0 & \text{otherwise.} \end{cases}$$

It is straightforward to check that $\bar{f}$ is a feasible solution to $\tau(I)$ and $\mu(\tau(I), \bar{f}) = |\mathcal{C}^*|$, hence $\mu(\tau(I)) \geq |\mathcal{C}^*| = \mu(I)$.

Conversely, for each feasible solution f' of $\tau(I)$ we construct a set packing $\mathcal{C}' \subseteq \mathcal{C}$ such that $|\mathcal{C}'| \geq \mu(\tau(I), f')$. Observe that all capacities and gains are integral, and so is the flow on each arc. First we may assume that $f'((v_j, t)) = 1$ provided $f'((v_j, t)) > 0$. Otherwise there would be an augmenting path s–v_j–t. Furthermore, we may assume that $f'((u_i, v_j)) = 1$ provided $f'((u_i, v_j)) > 0$. Otherwise for all (u_k, v_j) with $k \neq i$ and a positive flow $f'((u_k, v_j))$ we can shift the flow $f'((u_k, v_j))$ from the path s–u_k–v_j to the path s–u_i–v_j, and remove the flow on paths u_k–t if there is no flow on (s, u_k) after shifting. Clearly, the new flow is still feasible and the out-flow is not decreased. Hence, we have $f'((s, u_i)) = 1$ or 0 for all $1 \leq i \leq m$.

Now we define a subcollection

$$\mathcal{C}' = \{C_i \in \mathcal{C} \mid 1 \leq i \leq m \text{ and } f'((s, u_i)) = 1\}.$$

Then it is easily seen that $\mathcal{C}'$ is a set packing of S. Clearly, $\mu(\tau(I), f') \leq |\mathcal{C}'| \leq \mu(I)$, implying $\mu(\tau(I)) \leq \mu(I)$, $\mu(\tau(I)) = \mu(I)$, $\mu(I) - |C'| \leq \mu(\tau(I)) - \mu(\tau(I), f')$.

Therefore the approximation within a factor of n^ϵ is $\mathcal{NP}$-hard for some fixed $\epsilon > 0$. □

Theorem 10. *Approximating the minimum cost unit out-flow problem within a factor of n^ϵ in additive networks is $\mathcal{NP}$-hard for some $\epsilon > 0$, where n is an upper bound on the cost of the minimum cost unit out-flow.*

Proof. We slightly modify the proof of Theorem 9 by adding to the network N a new source s_0 and an arc (s_0, s) with capacity $u((s_0, s)) = 1$ and gain $g((s_0, s)) = p - 1$, and set the arc cost to

$$c(a) = \begin{cases} -1 & \text{if } e = (s, u_i), i = 1, \ldots, m, \\ 0 & \text{otherwise.} \end{cases}$$

Then it is easy to see that the MAX-SET-PACKING problem is equivalent to the minimum cost unit out-flow problem in the extended network with parameter p for $p = 2, \ldots, n + 1$, that is, determining the maximum p such that the unit out-flow problem is feasible. □

The $\mathcal{NP}$-hardness results indicate that the common network flow techniques with flow augmenting path will not work (efficiently) for computing a maximum flow. Unit-loss networks are an exception, where the maximum out-flow problem is solvable in polynomial time. Recall that in unit-loss networks the capacities are integral and so are the flows. Hence, a unit loss can be seen as a shortcut towards the sink and does not serve as a constraint.

Theorem 11. *Let N be a unit-loss network with n nodes and m arcs. The maximum out-flow problem can be solved in $\mathcal{O}(nm^2)$ time.*

Proof. Let N be a weakly unit-loss network with source s and sink t. Recall that the arcs have integral capacity and loss zero or one.

Construct a standard network N' by adding a new sink t' and an arc (t, t') with unbounded capacity. Split each lossy arc $a = (v, w)$ into an arc (v, v') with the same capacity as a and an arc (v', w) with capacity $u(a) - 1$, and add a "loss-arc" (v', t') with unit capacity.

Then use the Edmonds–Karp algorithm [7] for the computation of the maximal flow in N' from s to t'. The Edmonds–Karp algorithm computes flow augmenting paths according to their lengths, where the length is the number of arcs. Hence, at a split node v' it prefers the loss-arcs (v', t') and uses it first. In the residual network it will not use the backward of a loss-arc. If there were a flow augmenting path from s to t' using a backward arc (t', v'), then the path has a cycle at t', and the path is not a shortest path as taken by the Edmonds–Karp algorithm. The Edmonds–Karp algorithm computes a maximal flow, and the flow is integral at every arc.

In addition, all used loss-arcs are saturated, and all unused loss-arcs are unreachable from the source in the residual network of the maximum flow. This is due to the integrality of flow augmentations and the fact that Edmonds–Karp algorithm takes loss-arcs first at split nodes.

If f' is the maximal flow computed by the Edmonds–Karp algorithm in N', then the flow f with $f(a) = f'(a)$ if a is a flow conserving arc and $f(a) = f'(v, v') = f'(v', w) + 1$ if $a = (v, w)$ is a lossy arc in a maximum flow of N, which is split at v'. Clearly, $f_{out} \leq f'_{out}$. Let $a = (v, w)$ be a lossy arc. If $f'(a) > 0$, then $f'(a) \geq 1$ by the integrality and there is a unit flow on the loss-arc (v', t') by the Edmonds–Karp algorithm. Then $f(a) = f'(a) + 1$, and the flow into a in N equals the flow on a in N' plus the unit flow on the arc towards t'. Hence, $f_{out} \geq f'_{out}$. □

Acknowledgements

We would like to thank an anonymous referee for her/his valuable suggestions, which led to an improvement and to Christian Bachmaier and Michael Forster for their hints on an early version of Theorem 5. The second author's research is partially supported by a grant from Natural Science Foundation of China (No. 10171054). The paper has been selected from FAW 2009. (Handling editor: Xiaotie Deng.)

References

[1] R.K. Ahuja, T.L. Magnanti, J.B. Orlin, Network Flows, Prentice Hall, Englewood Cliffs, 1993.
[2] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties, Springer, Berlin, 1999.
[3] V. Batagelj, Personal communication, 1999.
[4] F.J. Brandenburg, Cycles in generalized networks, in: Proc. 28th Workshop Graph-Theoretic Concepts in Computer Science, WG 2002, in: Lecture Notes in Computer Science, vol. 2573, 2003, pp. 47–57. Corrigendum in Proc. 29th Workshop Graph-Theoretic Concepts in Computer Science, WG 2003, Lecture Notes in Computer Science Vol.
[5] B.V. Cherkassky, A.V. Goldberg, Negative-cycle detection algorithms, Math. Program. 85 (1999) 277–311.
[6] Cisco, IP Multicasting, Module 4 Basic Multicast Debugging, Cisco Systems Inc., 1998.
[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, 1990.
[8] G.B. Dantzig, Linear Programming and Extensions, Princeton Univ. Press, Princeton, NJ, 1963.
[9] L.K. Fleischer, K.D. Wayne, Fast and simple approximation schemes for generalized flow, Math. Program. 91 (2002) 215–238.
[10] L.R. Ford Jr., D.R. Fulkerson, Flows in Networks, Princeton Univ. Press, Princeton, NJ, 1962.
[11] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of $\mathcal{NP}$-Completeness, W.H. Freeman, San Francisco, 1979.
[12] A.V. Goldberg, S.A. Plotkin, E. Tardos, Combinatorial algorithms for the generalized circulation problem, Math. Oper. Res. 16 (1991) 351–381.
[13] D. Goldfarb, Z. Jin, J. Orlin, Polynomial-time highest-gain augmenting path algorithms for the generalized circulation problem, Math. Oper. Res. 22 (1997) 793–802.
[14] D. Goldfarb, Z. Jin, Y. Lin, A polynomial dual simplex algorithm for the generalized circulation problem, Math. Progam., Ser. A 91 (2002) 271–288.
[15] S. Haider, Passau und der Salzhandel nach Österreich, in [29], pp. 221–236.
[16] D. Hochbaum, Approximation Algorithms for $\mathcal{NP}$-hard Problems, PWS, Boston, 1997.
[17] L.V. Kantorovich, Mathematical methods of organizing and planning production, Publication House of the Leningrad State University, (1939), 68. Translated in Management Science 6 (1960) 366–422.
[18] N. Karmarkar, A new polynomial-time algorithm for linear programming, Combinatorica 4 (1984) 373–395.
[19] W.S. Jewell, Optimal flow through networks with gains, Oper. Res. 10 (1962) 476–499.
[20] V. Kann, Maximum bounded 3-dimensional matching is MAX-SNP-complete, Inform. Proc. Lett. 37 (1991) 27–35.
[21] L.G. Khachian, Polynomial algorithms in linear programming, Zhurnal Vychislitel. Mat. Mat. Fiziki 20 (1980) 53–72.
[22] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehard, and Winston, New York, 1976.
[23] K. Mehlhorn, S. Näher, LEDA: A Platform for Combinatorial and Geometric Computing, Cambridge University Press, Cambridge, 1999.
[24] P. Milgrom, J. Roberts, Economics, Organization, and Management, Prentice-Hall, Englewood Cliffs, 1992.
[25] J.D. Oldham, Combinatorial approximation algorithms for generalized flow problems, J. Algorithms 38 (2001) 135–168.
[26] R.E. Tarjan, Data Structures and Network Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, 1983.
[27] K.D. Wayne, A polynomial combinatorial algorithm for generalized cost flow, Math. Oper. Res. 27 (2002) 445–459.
[28] http://wienerboerse.at/glossary/1/19/446.
[29] H.W. Wurster, M. Brunner, R. Loibl, and A. Brunner (Eds.), Weisses Gold: Passau Vom Reichtum einer europäischen Stadt, Katalog zur Ausstellung von Stadt und Diözese Passau im Oberhausmuseum Passau, 1995.
[30] D. Zuckerman, On unapproximable versions of $\mathcal{NP}$-complete problems, SIAM J. Comput. 26 (1996) 1293–1304.