Complexity and Approximability of the Maximum Flow Problem with Minimum Quantities

Clemens Thielen[1] and Stephan Westphal[2]

[1] Department of Mathematics, University of Kaiserslautern
Paul-Ehrlich-Str. 14, D-67663 Kaiserslautern, Germany
thielen@mathematik.uni-kl.de

[2] University of Goettingen, Institute for Numerical and Applied Mathematics,
Lotzestr. 16-18, D-37083 Goettingen, Germany
s.westphal@math.uni-goettingen.de

Abstract. We consider the *maximum flow problem with minimum quantities (MFPMQ)*, which is a variant of the maximum flow problem where the flow on each arc in the network is restricted to be either zero or above a given lower bound (a *minimum quantity*), which may depend on the arc. This problem has recently been shown to be weakly NP-complete even on series-parallel graphs.

In this paper, we provide further complexity and approximability results for MFPMQ and several special cases. We first show that it is strongly NP-hard to approximate MFPMQ on general graphs (and even bipartite graphs) within any positive factor. On series-parallel graphs, however, we present a pseudo-polynomial time dynamic programming algorithm for the problem.
We then study the case that the minimum quantity is the same for each arc in the network and show that, under this restriction, the problem is still weakly NP-complete on general graphs, but can be solved in strongly polynomial time on series-parallel graphs. On general graphs, we present a $(2 - \frac{1}{\lambda})$-approximation algorithm for this case, where λ denotes the common minimum quantity of all arcs.

Keywords: maximum flow problem, minimum quantities, computational complexity, approximation algorithms

1 Introduction

The maximum flow problem is a classical problem in network flow theory (cf., for example, [1]), which can be used to model many problems in logistics and transportation.

In this paper, we consider a variant of the problem where the flow on each arc in the network is restricted to be either zero or above a given lower bound (a *minimum quantity*), which may depend on the arc. This problem is motivated by pipe systems where, for example in a wastewater system, a used pipe will get clogged if not at least a minimum amount of water runs through it. Other applications stem from minimum lot sizes in transportation and logistics (cf. [4]). Formally, the maximum flow problem with minimum quantities (MFPMQ) is defined as follows:

Definition 1 (Maximum Flow Problem with Minimum Quantities).
INSTANCE: *A directed graph $G = (V, R)$, a source $s \in V$, a sink $t \in V$, arc capacities $u : R \to \mathbb{N} \cup \{+\infty\}$, and minimum quantities $\lambda : R \to \mathbb{N}_0$ such that $\lambda(r) \leq u(r)$ for all $r \in R$.*
TASK: *Find a feasible s-t-flow $f : R \to \mathbb{N}_0$ of maximum flow value in G.*

In the above definition, a *feasible s-t-flow* is a function $f : R \to \mathbb{N}_0$ that satisfies the flow conservation constraints and respects the minimum quantities and upper capacities on the arcs. Denoting the set of arcs starting in $v \in V$ by $\delta^+(v)$ and the set of arcs ending in v by $\delta^-(v)$, these requirements can be expressed by the following constraints:

$$\begin{aligned} \sum_{r \in \delta^+(v)} f(r) - \sum_{r \in \delta^-(v)} f(r) = 0 & \qquad \forall\, v \in V \setminus \{s, t\}, \\ f(r) \in \{0\} \cup [\lambda(r), u(r)] & \qquad \forall\, r \in R. \end{aligned}$$

The *flow value* of f is then defined as $\text{val}(f) := \sum_{r \in \delta^+(s)} f(r) - \sum_{r \in \delta^-(s)} f(r)$.

In the decision version of the problem, a bound $F \in \mathbb{N}$ on the flow value is given and the question is whether there exists a feasible flow with flow value at least F. When stating results about the computational complexity of an optimization problem such as MFPMQ, we will always mean the complexity of the corresponding decision problem.

1.1 Previous Work

The classical maximum flow problem is well-studied in literature. A thorough introduction to the problem is given in [1]. The maximum flow problem with minimum quantities was introduced in [4], where the problem was shown to be weakly NP-complete even on series-parallel graphs and Lagrangean relaxation techniques and heuristics for solving the problem were studied.

Minimum quantities have also been studied recently for minimum cost network flow problems [5, 7, 9], where introducing minimum quantities results in the minimum cost flow problem becoming strongly NP-complete [5]. Moreover, it was shown in [5] that (unless P = NP) no polynomial time $g(|I|)$-approximation for the minimum cost flow problem with minimum quantities exists for any polynomially computable function $g : \mathbb{N}_+ \to \mathbb{N}_+$, where $|I|$ denotes the encoding length of the given instance[1].

[1] In addition, a dynamic programming algorithm and fully polynomial time approximation scheme (FPTAS) for the minimum cost flow problem with minimum quantities on series-parallel graphs were presented in [5]. These results, however, were shown to be erroneous in a recent erratum [6].

1.2 Our Contribution

We show that it is strongly NP-hard to approximate MFPMQ within any positive factor even when the graph is bipartite. On series-parallel graphs, however, we present a pseudo-polynomial time dynamic programming algorithm for the problem. This result is best possible for series-parallel graphs since the proof of weak NP-completeness of MFPMQ on series-parallel graphs given in [4] in fact shows weak NP-hardness of approximation. Moreover, our result shows that the problem cannot be strongly NP-hard for series-parallel graphs (unless P = NP).

We then consider the case that the minimum quantity is the same for each arc in the network and show that, on general graphs, the problem is still weakly NP-complete under this assumption. Moreover, we present a $(2 - \frac{1}{\lambda})$-approximation algorithm for this case, where $\lambda \geq 2$ denotes the common minimum quantity of all arcs (for $\lambda \in \{0, 1\}$, the problem can be solved optimally in strongly polynomial time as a standard maximum flow problem without minimum quantities). Finally, we show that the problem on series-parallel graphs can be solved in strongly polynomial time when the minimum quantity is the same for each arc in the network.

2 Hardness of Approximation on Bipartite Graphs

We start our discussion by showing that it is strongly NP-hard to approximate MFPMQ within any positive factor even when the graph is bipartite. For the proof, we use a reduction from the well-known NP-complete exact cover by 3-sets problem (cf. [3]). An instance of EXACT COVER 3-SETS consists of a finite set $X = \{x_1, \ldots, x_n\}$ with $n = 3q,\ q \in \mathbb{N}$, and a collection of 3-element subsets $S_1, \ldots, S_m \subseteq X$ and the question is whether there exists an exact cover of X by some of the subsets $S_1, \ldots, S_m$, i.e., indices $c_1, \ldots, c_q$ such that each element $x_i \in X$ is contained in exactly one of the sets $S_{c_1}, \ldots, S_{c_q}$.

Theorem 1. *MFPMQ is strongly NP-hard to approximate within any positive factor even when the graph is bipartite.*

Proof. Given an instance of EXACT COVER 3-SETS specified by the set $X = \{x_1, \ldots, x_n\}$ and the 3-element subsets $S_1, \ldots, S_m \subseteq X$, we construct an instance of MFPMQ such that any positive factor approximation for this instance can be used to decide whether the given instance of EXACT COVER 3-SETS has a solution: The graph $G = (V, R)$ contains a source s, a sink t, a vertex x_i for every element $x_i \in X$, and a vertex S_j for each subset. Moreover, there is one additional vertex s'. There are arcs from the source s to s', from s' to each set S_j, and from each element x_i to the sink t. Finally, each set S_j has arcs to all elements x_i contained in it.

The arc capacities are defined as follows: The arc (s, s') has capacity $u(s, s') = n$. Each arc $r = (s', S_j)$ connecting s' to a subset S_j has capacity $u(r) = |S_j| = 3$. All arcs (x_i, t) connecting an element x_i to the sink t and all arcs (S_j, x_i) connecting a subset to an element have capacity 1. Nontrivial minimum quantities are only imposed on the arc (s, s'), for which $\lambda(s, s') = n$, and on the arcs connecting s' to the subsets S_j, where we have $\lambda(s', S_j) = |S_j| = 3$. For all other arcs $r \in R$, we set $\lambda(r) := 0$.

Figure 1 shows the graph G for an example instance. Note that G is not series-parallel as the subgraph induced by s', S_2, S_3, x_3, x_4, and t is homeomorphic to Braess's graph (the forbidden minor for series-parallel graphs).

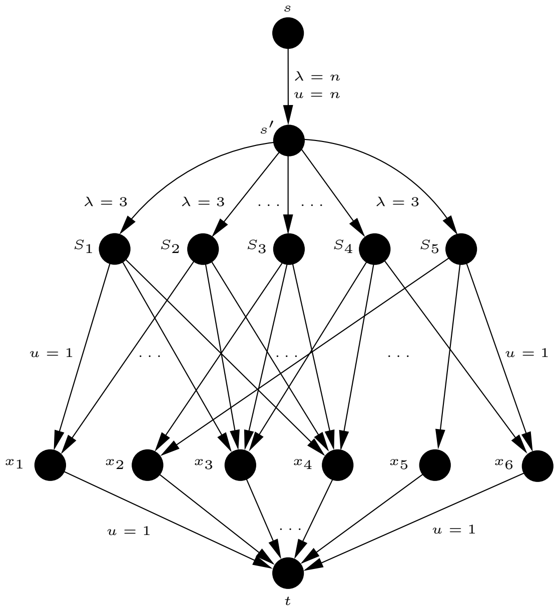

Fig. 1. The graph G for the set $X = \{x_1, \dots, x_6\}$ (so $n = 6$) and the subsets $S_1 = S_2 = \{x_1, x_3, x_4\}, S_3 = \{x_2, x_3, x_4\}, S_4 = \{x_3, x_4, x_6\}, S_5 = \{x_2, x_5, x_6\}$.

We now show that there exists an exact cover of $X = \{x_1, \dots, x_n\}$ by some of the subsets $S_1, \dots, S_m$ if and only if there exists a feasible s-t-flow f of flow value n in G. Since the minimum quantity and capacity of n for the arc (s, s') imply that any feasible flow in G must have value n or 0 and G is bipartite, this will prove the claim.

First suppose that there exists a cover $X = S_{c_1} \cup S_{c_2} \cup \dots \cup S_{c_q}$ of X. Then we define a flow f in G as follows: We set $f(s, s') = n$, $f(s', S_{c_j}) = 3$ for all $j \in \{1, \dots, q\}$, $f(S_{c_j}, x_i) = 1$ if and only if $x_i \in S_{c_j}$, and $f(x_i, t) = 1$ for all $x_i \in X$. On all other arcs, the flow is zero. Feasibility of f now follows immediately from the fact that the sets S_{c_j} form an exact cover of X. Since the flow value of f is $f(s, s') = n$, this shows that f is as desired.

Conversely suppose that there is a feasible flow f of flow value n in G. Then it follows that $f(x_i, t) = 1$ for all $x_i \in X$, so each element is covered exactly once by the sets S_j with $f(s', S_j) = 3$, which shows that an exact cover exists.

3 A Dynamic Programming Algorithm for Series-Parallel Graphs

In this section, we present a pseudo-polynomial time dynamic programming algorithm that solves MFPMQ exactly on series-parallel graphs.

A directed graph with two distinguished vertices s (the *source*) and t (the *sink*) is *series-parallel* if it is obtained from a set of copies of the directed one-arc graph K_2 by a finite sequence of series compositions (identifying the sink of the first graph with the source of the second graph) and parallel compositions (identifying the sources and sinks of the two graphs, respectively). For further details on series-parallel graphs, we refer to [2].

For a given series-parallel graph $G = (V, R)$ with $|V| =: n$ vertices, $|R| =: m$ arcs, source s and sink t, arc capacities $u : R \to \mathbb{N}$, and minimum quantities $\lambda : R \to \mathbb{N}$, we denote the maximum value of a feasible s-t-flow that is less or equal to a given integer $M \in \mathbb{N}_0$ by $F_G(M)$. Starting from the leaves of the decomposition tree of the series-parallel graph, our pseudo-polynomial time algorithm computes the value $F_G(M)$ for a suitable upper bound M on the maximum flow value recursively by dynamic programming:

If G consists of only one arc r, the value $F_G(\gamma)$ can be computed in constant time for each $\gamma \in \{0, \dots, M\}$ as

$$F_G(\gamma) = \min\{\gamma, u(r)\}$$

or $F_G(\gamma) = 0$ if this value is smaller than the minimum quantity $\lambda(r)$ of the arc.

For the case that G is the series or parallel composition of G_1 and G_2, the value $F_G(\gamma)$ can be computed as stated in the following lemmas:

Lemma 1. *If G is the series composition of G_1 and G_2, then*

$$F_G(\gamma) = \max\{F_{G_1}(j) : 0 \leq j \leq \gamma \text{ and } F_{G_1}(j) = F_{G_2}(F_{G_1}(j))\}.$$

Proof. "$\geq$": If $F_{G_1}(j) = F_{G_2}(F_{G_1}(j))$ for some $j \in \{0, \dots, \gamma\}$, then, by definition of $F_{G_1}(j)$, it follows that $F_{G_1}(j)$ units of flow can be sent through G_1 and, by the equality, these $F_{G_1}(j)$ units can also be sent through G_2. Hence, there exists a feasible s-t-flow in G of value $F_{G_1}(j)$, which proves that $F_G(\gamma) \geq F_{G_1}(j)$.

"$\leq$": Every feasible s-t-flow f in G with flow value $0 \leq \text{val}(f) \leq \gamma$ must send $\text{val}(f)$ units of flow through G_1, so we must have $\text{val}(f) = F_{G_1}(\text{val}(f))$. Moreover, since this amount of flow must also be sent through G_2, we obtain that also $F_{G_2}(F_{G_1}(\text{val}(f))) = F_{G_2}(\text{val}(f)) = \text{val}(f)$, which proves that $\text{val}(f) = F_{G_1}(\text{val}(f)) = F_{G_2}(F_{G_1}(\text{val}(f)))$. Hence, the maximum on the right hand side is at least $\text{val}(f)$.

Lemma 2. *If G is the parallel composition of G_1 and G_2, then*

$$F_G(\gamma) = \max_{0 \leq j \leq \gamma} F_{G_1}(j) + F_{G_2}(\gamma - j).$$

Proof. "$\geq$": Trivial.

"$\leq$": Let f be a feasible s-t-flow in G with flow value $0 \leq \mathrm{val}(f) \leq \gamma$ that sends $\mathrm{val}_1(f)$, $\mathrm{val}_2(f)$ units of flow through G_1 and G_2, respectively. Then we have $\mathrm{val}_1(f) = F_{G_1}(\mathrm{val}_1(f))$ and $\mathrm{val}_1(f) + \mathrm{val}_2(f) = \mathrm{val}(f) \leq \gamma$, so $\mathrm{val}_2(f) \leq \gamma - \mathrm{val}_1(f)$. Consequently, we obtain that $\mathrm{val}_2(f) = F_{G_2}(\mathrm{val}_2(f)) \leq F_{G_2}(\gamma - \mathrm{val}_1(f))$. Hence:

$$\mathrm{val}(f) = \mathrm{val}_1(f) + \mathrm{val}_2(f) \leq F_{G_1}(\mathrm{val}_1(f)) + F_{G_2}(\gamma - \mathrm{val}_1(f)).$$

As $0 \leq \mathrm{val}_1(f) \leq \gamma$, this shows that the maximum on the right hand side is at least $F_{G_1}(\mathrm{val}_1(f)) + F_{G_2}(\gamma - \mathrm{val}_1(f)) \geq \mathrm{val}(f)$.

Using the recursions stated in Lemma 1 and 2, the value $F_G(\gamma)$ can be computed in time $\mathcal{O}(\gamma)$ when the values $F_{G_k}(j)$ for $k \in \{1, 2\}, j \in \{0, \ldots, \gamma\}$ are already known. Thus, we need time $\mathcal{O}(\sum_{\gamma=0}^{M} \gamma) = \mathcal{O}(M^2)$ to compute the values $F_{G'}(\gamma)$, $\gamma = 0, \ldots, M$, for each series-parallel graph G' representing a branch vertex of the decomposition tree of the given series-parallel graph. For a leaf of the decomposition tree, we only need constant time to compute each value, which yields a total time requirement of $\mathcal{O}(M)$ for computing all values $F_{G'}(\gamma)$, $\gamma = 0, \ldots, M$, if G' is a leaf. Hence, since the decomposition tree of a series-parallel graph with n vertices and m arcs has $\mathcal{O}(m)$ vertices and can be computed in $\mathcal{O}(m)$ time if the graph is connected (cf. [8]), our procedure computes the value $F_G(M)$ in $\mathcal{O}(m \cdot M^2)$ time.

To obtain an optimal solution of MFPMQ, the algorithm computes the value $F_G(M)$ for a suitable upper bound M on the maximum flow value in G. For example, we can set

$$M := \sum_{r \in \delta^+(s)} u(r)$$

to be the sum of all capacities of the arcs starting in the source. Hence, our algorithm finds an optimal solution of MFPMQ in time $\mathcal{O}(m \cdot M^2)$. Note that this running time bound is only pseudo-polynomial since it depends on an upper bound M on the maximum flow value (and, hence, on the arc capacities).

4 Identical Minimum Quantities on General Graphs

We now show that MFPMQ on general graphs is weakly NP-complete even if we restrict ourselves to instances with identical minimum quantities on all arcs. We show, however, how a $(2 - \frac{1}{\lambda})$-approximation for the case of identical minimum quantities can be obtained in polynomial time, where $\lambda \geq 2$ denotes the common minimum quantity of all arcs (for $\lambda \in \{0, 1\}$, the problem can be solved optimally in strongly polynomial time as a standard maximum flow problem without minimum quantities).

We start by showing weak NP-completeness. To show NP-hardness, we use a reduction from the (weakly) NP-complete partition problem (cf. [3]). An instance of PARTITION consists of n positive integers $a_1, \ldots, a_n \in \mathbb{N}$ and the question is whether there exists a subset S of $\{1, \ldots, n\}$ such that $\sum_{i \in S} a_i = \frac{1}{2} \sum_{i=1}^{n} a_i$ (in which case we call $(S, \{1, \ldots, n\} \setminus S)$ a *partition* of $a_1, \ldots, a_n$).

Theorem 2. *MFPMQ with identical minimum quantities is weakly* NP*-complete.*

Proof. Membership in NP is obvious as feasibility and flow value of a given flow f can be checked easily in polynomial time. To show NP-hardness, we reduce PARTITION to MFPMQ with identical minimum quantities in polynomial time. Given an instance of PARTITION specified by $a_1, \ldots, a_n$, first note that we may assume without loss of generality that there does not exist $j \in \{1, \ldots, n\}$ such that $a_j \geq \frac{1}{2} \cdot \sum_{i=1}^{n} a_i$ as, for $a_j > \frac{1}{2} \cdot \sum_{i=1}^{n} a_i$, the instance can clearly not have a solution and, for $a_j = \frac{1}{2} \cdot \sum_{i=1}^{n} a_i$, the instance has the trivial solution given by the subset $S = \{j\}$.

We now construct an instance of MFPMQ with identical minimum quantities as follows: The graph $G = (V, R)$ contains a source s, a sink t, a vertex v_i for every $i \in \{1, \ldots, n\}$, and vertices x_1, x_2, y_1, y_2. There are arcs from the source s to each v_i with capacity $\lambda + a_i$, and from each v_i to each x_j with capacity $\lambda + a_i$, where $\lambda := \frac{1}{2} \cdot \sum_{i=1}^{n} a_i$ is the minimum quantity on all arcs. Furthermore, there are $n + 1$ parallel arcs with capacity λ from x_j to y_j and one arc from y_j to t with capacity $(n + 1) \cdot \lambda$ for $j = 1, 2$. Figure 2 shows the graph G for the case $n = 5$.

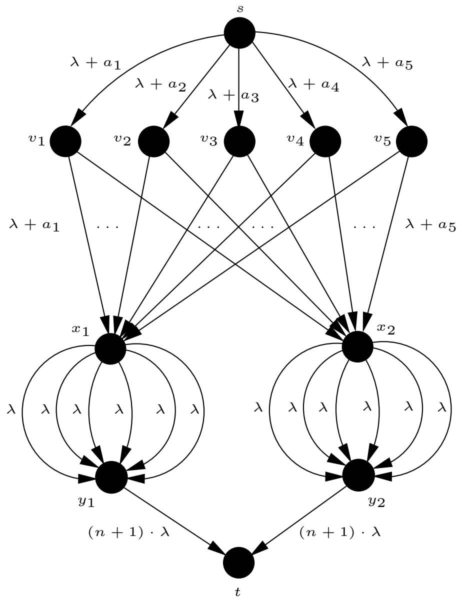

Fig. 2. The graph G from the proof of Theorem 2 for $n = 5$. The minimum quantity on all arcs is λ and the capacities are given by the arc labels.

We now show that there exists a partition (S_1, S_2) of $a_1, \dots, a_n$ if and only if the maximum flow value in G is equal to $n \cdot \lambda + \sum_{i=1}^{n} a_i = (n+2) \cdot \lambda$.

First suppose that there exists a partition (S_1, S_2) of $a_1, \dots, a_n$. Then we define a flow f in G as follows: We set $f(s, v_i) = \lambda + a_i$ for all $i \in \{1, \dots, n\}$, $f(v_i, x_j) = \lambda + a_i$ for all $j \in \{1, 2\}, i \in S_j$. For $i \notin S_j$, we set $f(v_i, x_j) = 0$. From the $n+1$ parallel arcs leading from x_j to y_j, the flow f has value λ on $|S_j| + 1$ of them and value 0 on the others. Finally, the flow on the arcs (y_1, t) and (y_2, t) attains values $(|S_1| + 1) \cdot \lambda$ and $(|S_2| + 1) \cdot \lambda$, respectively. It is then clear that f is a feasible s-t-flow with value $(n+2) \cdot \lambda$

Conversely, suppose that there exists a feasible flow f of flow value $n \cdot \lambda + \sum_{i=1}^{n} a_i = (n+2) \cdot \lambda$ in G. Then it follows that $f(s, v_i) = \lambda + a_i$ for all $i \in \{1, \dots, n\}$. As there is a minimum quantity of λ for each of the arcs (v_i, x_j) and $\lambda > a_i$ by our assumption on the PARTITION instance, exactly one of the arcs $(v_i, x_1), (v_i, x_2)$ can have a positive flow value for each $i \in \{1, \dots, n\}$, and this value has to be $\lambda + a_i$ in order to fulfill the flow conservation constraints. For $j = 1, 2$, let S_j be the set of all i such that $f(v_i, x_j) = \lambda + a_i$. Observe that (S_1, S_2) is a partition of $\{1, \dots, n\}$. The parallel arcs between x_j and y_j ensure that the total flow going into x_j has to be an integer multiple of λ. Since $\sum_{i=1}^{n} a_i = 2\lambda$, we have $\sum_{i \in S_1} a_i \in \{0, \lambda, 2\lambda\}$. As the total capacity of the arcs going from x_1 to y_1 is $(n+1) \cdot \lambda$, it is impossible that $\sum_{i \in S_1} a_i = 2\lambda$. As $\sum_{i \in S_1} a_i = 0$ would result in $\sum_{i \in S_2} a_i = 2\lambda$, the only possible case is that $\sum_{i \in S_1} a_i = \lambda = \frac{1}{2} \cdot \sum_{i=1} a_i$. This shows that (S_1, S_2) is a solution to the given instance of PARTITION, which proves NP-completeness.

We now show how a $(2 - \frac{1}{\lambda})$-approximation for MFPMQ with identical minimum quantity $\lambda \geq 2$ on all arcs can be obtained in polynomial time. Our algorithm can be stated as follows:

Algorithm 3 (MFPMQ with Identical Minimum Quantities).

INPUT: A directed graph $G = (V, R)$, a source $s \in V$, a sink $t \in V$, arc capacities $u : R \to \mathbb{N}$, and an integer minimum quantity $\lambda \geq 2$ such that $u(r) \geq \lambda$ for all $r \in R$.

OUTPUT: A feasible s-t-flow $f : R \to \mathbb{N}_0$.

METHOD: Define new arc capacities $\bar{u}$ by setting $\bar{u}(r) := \lfloor \frac{u(r)}{\lambda} \rfloor$, compute an integral maximum s-t-flow $\bar{f}$ in G without minimum quantities with respect to these arc capacities, and return $f := \lambda \cdot \bar{f}$.

Observe that, since the arc capacities $\bar{u}$ are integral, an integral maximum flow $\bar{f}$ in G without minimum quantities exists and can be computed in strongly polynomial time (cf. [1]). Moreover, the flow f returned by Algorithm 3 is feasible: Since $\bar{f}$ satisfies the flow conservation constraints at each node, also $f = \lambda \cdot \bar{f}$ does so. Moreover, we have

$$f(r) = \lambda \cdot \bar{f}(r) \leq \lambda \cdot \bar{u}(r) = \lambda \cdot \left\lfloor \frac{u(r)}{\lambda} \right\rfloor \leq u(r)$$

for each arc $r \in R$, so f respects the arc capacities, and since the flow on each arc that has positive flow under $\bar{f}$ is at least one, the flow on each arc under $f = \lambda \cdot \bar{f}$ is either zero or at least λ, so f also respects the minimum quantities.

In order to prove that Algorithm 3 is a $(2 - \frac{1}{\lambda})$-approximation for MFPMQ with identical minimum quantities, we need the following lemma:

Lemma 3. *For all integers $u \geq \lambda \geq 2$:*

$$\lfloor \frac{u}{\lambda} \rfloor \geq \frac{\lambda}{2\lambda - 1} \cdot \frac{u}{\lambda}$$

Proof. Writing $u = k \cdot \lambda + q$ for integers $0 \leq q \leq \lambda - 1$ and $k \geq 1$, we obtain that

$$\frac{\lfloor \frac{u}{\lambda} \rfloor}{\frac{u}{\lambda}} = \frac{k}{k + \frac{q}{\lambda}} \geq \frac{k}{k + \frac{\lambda - 1}{\lambda}} = \frac{k\lambda}{(k+1)\lambda - 1}. \tag{1}$$

The derivative of this term with respect to k is

$$\frac{\lambda((k+1)\lambda - 1) + k\lambda^2}{((k+1)\lambda - 1)^2} \geq 0,$$

so (1) is minimized for $k = 1$. Hence, we obtain that

$$\frac{\lfloor \frac{u}{\lambda} \rfloor}{\frac{u}{\lambda}} \geq \frac{\lambda}{2\lambda - 1},$$

which proves the claim.

Now let (S, T) be a minimum s-t-cut in G with respect to the capacities $\bar{u}$ and let $\delta^+(S)$ denote the set of arcs with start point in S and end point in T. Then, by the Max-Flow Min-Cut Theorem (cf., for example, [1]) and Lemma 3, the flow value of $\bar{f}$ satisfies

$$\begin{aligned} \mathsf{val}(\bar{f}) &= \sum_{r \in \delta^+(S)} \bar{u}(r) = \sum_{r \in \delta^+(S)} \left\lfloor \frac{u(r)}{\lambda} \right\rfloor \geq \sum_{r \in \delta^+(S)} \frac{\lambda}{2\lambda - 1} \cdot \frac{u(r)}{\lambda} \\ &= \frac{1}{2\lambda - 1} \cdot \sum_{r \in \delta^+(S)} u(r) \geq \frac{1}{2\lambda - 1} \cdot \text{OPT}, \end{aligned}$$

where OPT denotes the maximum flow value for the original instance. Thus, we obtain that $\mathsf{val}(f) = \lambda \cdot \mathsf{val}(\bar{f}) \geq \frac{\lambda}{2\lambda - 1} \cdot \text{OPT}$, which proves the following theorem:

Theorem 4. *Algorithm 3 is a $(2 - \frac{1}{\lambda})$-approximation for MFPMQ with identical minimum quantity $\lambda \geq 2$ on all arcs.*

Note that the above analysis of Algorithm 3 is tight for every $\lambda \geq 2$: If the graph G consists of only a single arc (s, t) with minimum quantity λ and capacity $2\lambda - 1$, Algorithm 3 returns a flow f of value $\mathsf{val}(f) = \lambda$, while the optimal solution has flow value $2\lambda - 1$.

Also note that, since we only used that OPT is bounded from above by the capacity of the cut (S, T), we can also conclude that $\mathsf{val}(f)$ is at least $\frac{\lambda}{2\lambda-1}$ times the value of a maximum s-t-flow in G *without minimum quantities*. Hence, since f is a feasible flow for the instance with minimum quantities, we obtain the following corollary:

Corollary 1. *Let $\mathsf{OPT}_{\mathsf{MQ}}$ denote the maximum value of an s-t-flow for a given instance of MFPMQ with identical minimum quantity $\lambda \geq 2$ on all arcs and let OPT denote the maximum value of an s-t-flow for the same instance without minimum quantities. Then $\mathsf{OPT}_{\mathsf{MQ}} \geq \frac{\lambda}{2\lambda-1} \cdot \mathsf{OPT}$.*

Note that, in the case of *non-identical* minimum quantities on the arcs, no universal upper bound on the ratio $\mathsf{OPT}/\mathsf{OPT}_{\mathsf{MQ}}$ exists, even for series-parallel graphs: If G is the series composition of two single arcs $r_1 = (s, v)$ and $r_2 = (v, t)$ with $\lambda(r_1) = u(r_1) = 1$ and $\lambda(r_2) = u(r_2) = 2$, the maximum flow value without minimum quantities is one, whereas only the zero flow is feasible for the instance with minimum quantities.

5 Identical Minimum Quantities on Series-Parallel Graphs

While MFPMQ is (weakly) NP-hard to solve on series-parallel graphs (as shown in [4]) as well as when the minimum quantity on all arcs is identical (as shown in the previous section), we now show that the problem can be solved in strongly polynomial time when both restrictions are present, i.e., in the case of identical minimum quantities on series-parallel graphs. To do so, we present a strongly polynomial time dynamic programming algorithm for this case.

For a given series-parallel graph $G = (V, R)$ with $|V| =: n$ vertices, $|R| =: m$ arcs, source s and sink t, arc capacities $u : R \to \mathbb{N}$, and an identical minimum quantity $\lambda \geq 2$ on all arcs and an integer $k \in \{0, \ldots, m\}$, we denote the set of all flow values of feasible s-t-flows f in G such that $f(r) > 0$ (and, thus, $f(r) \geq \lambda$) for exactly k arcs in G by $S_G(k)$. Starting from the leaves of the decomposition tree of the series-parallel graph, our algorithm computes the sets $S_G(k)$ recursively by dynamic programming:

If G consists of only one arc r, the set $S_G(k)$ can be computed in constant time for each $k \in \{0, \ldots, m\}$ as

$$S_G(k) = \begin{cases} \{0\} & \text{for } k = 0 \\ [\lambda, u(r)] & \text{for } k = 1 \\ \emptyset & \text{for } k > 1 \end{cases} \tag{2}$$

If G is the series composition of G_1 and G_2, we have

$$S_G(k) = \bigcup_{\substack{0 \le k_1, k_2 \le k \\ k_1 + k_2 = k}} S_{G_1}(k_1) \cap S_{G_2}(k_2), \tag{3}$$

and if G is the parallel composition of G_1 and G_2, we have

$$S_G(k) = \bigcup_{\substack{0 \le k_1, k_2 \le k \\ k_1 + k_2 = k}} S_{G_1}(k_1) + S_{G_2}(k_2), \tag{4}$$

where $S_{G_1}(k_1) + S_{G_2}(k_2) := \{F_1 + F_2 : F_1 \in S_{G_1}(k_1), F_2 \in S_{G_2}(k_2)\}$ denotes the Minkowski sum of $S_{G_1}(k_1)$ and $S_{G_2}(k_2)$.

Using these recursions and the decomposition tree of the given series-parallel graph, we can recursively compute the sets $S_G(k)$ for $k = 0, \ldots, m$ and the maximum flow value of the instance is then given as

$$\max_{k=0,\ldots,m} \max(S_G(k)).$$

In order to obtain a polynomial running time, however, we have to show that the sets $S_G(k)$ can be represented efficiently. To do so, we use the following observation, which follows directly from (2) - (4) and the fact that the union, intersection, and Minkowski sum of intervals starting at integer multiples of λ again consist only of intervals starting at integer multiples of λ:

Observation 5. If $G = (V, R)$ is a series-parallel graph with $m := |R|$ arcs and $k \in \{0, \ldots, m\}$, then each set $S_G(k)$ can be written as

$$S_G(k) = \cup_{l \in I} [l \cdot \lambda, U_G(l)],$$

where $I \subseteq \{0, \ldots, m\}$, $l \cdot \lambda \le U_G(l) \le (l+1) \cdot \lambda$ for $l \le m - 1$, and $m \cdot \lambda \le U_G(m) \le \sum_{r \in R} u(r)$.

By Observation 5, each set $S_G(k)$ can be represented by at most $(m+1)$ intervals whose endpoints are of polynomial encoding length. Using the recursions (3) and (4), we can, thus, compute the representation of the set $S_G(k)$ in time $\mathcal{O}(k \cdot (m+1)^2) = \mathcal{O}(k \cdot m^2)$ when the sets $S_{G_1}(k_1), S_{G_2}(k_2)$ for $0 \le k_1, k_2 \le k$ are already known. Hence, we need time $\mathcal{O}(\sum_{k=0}^{m} k \cdot m^2) = \mathcal{O}(m^4)$ to compute the sets $S_k(G')$, $k = 0, \ldots, m$, for each series-parallel graph G' representing a branch vertex of the decomposition tree of the given series-parallel graph. For a leaf vertex of the decomposition tree, we only need constant time for each set, which yields a total time requirement of $\mathcal{O}(m)$ for computing all sets $S_k(G')$, $k = 0, \ldots, m$ if G' is a leaf. Since the decomposition tree of a connected, series-parallel graph G with m arcs has $\mathcal{O}(m)$ vertices, this implies that our procedure computes all sets $S_G(k)$ for $k = 0, \dots, m$ in $\mathcal{O}(m \cdot m^4) = \mathcal{O}(m^5)$ time. Moreover, since each set $S_G(k)$ is represented by at most $m+1$ intervals, we can afterwards compute the maximum flow value of the instance given as

$$\max_{k=0,\dots,m} \max(S_G(k))$$

in $\mathcal{O}(m \cdot (m+1)) = \mathcal{O}(m^2)$ time, so we obtain a total running time of $\mathcal{O}(m^5) + \mathcal{O}(m^2) = \mathcal{O}(m^5)$, which proves the following theorem:

Theorem 6. *An optimal solution of an instance of MFPMQ with identical minimum quantities on series-parallel graphs can be computed in $\mathcal{O}(m^5)$ time.*

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin: *Network Flows.* Prentice Hall (1993).

[2] A. Brandstädt, V. B. Le, and J. P. Spinrad: *Graph Classes: A Survey.* SIAM Monographs on Discrete Mathematics and Applications (1999).

[3] M. R. Garey and D. S. Johnson: *Computers and Intractability (A Guide to the Theory of NP-Completeness).* W.H. Freeman and Company, New York (1979).

[4] D. Haugland, M. Eleyat, and M. L. Hetland: *The maximum flow problem with minimum lot sizes.* In: *Proceedings of the 2nd International Conference on Computational Logistics (ICCL)*, volume 6971 of *LNCS.* pp. 170–182 (2011).

[5] S. O. Krumke and C. Thielen: *Minimum cost flows with minimum quantities.* Information Processing Letters 111(11): pp. 533–537 (2011).

[6] S. O. Krumke and C. Thielen: *Erratum to "Minimum cost flows with minimum quantities" [5].* Information Processing Letters 112(13): pp. 523–524 (2012).

[7] H. G. Seedig: *Network flow optimization with minimum quantities.* In: *Operations Research Proceedings 2010: Selected Papers of the Annual International Conference of the German Operations Research Society.* Springer, pp. 295–300 (2011).

[8] J. Valdes, R. E. Tarjan, and E. L. Lawler: *The recognition of series parallel digraphs.* SIAM Journal on Computing 11(2): pp. 298–313 (1982).

[9] X. Zhu, Q. Yuan, A. Garcia-Diaz, and L. Dong: *Minimal-cost network flow problems with variable lower bounds on arc flows.* Computers and Operations Research 38(8): pp. 1210–1218 (2011).