APPLIED MATHEMATICS

HELMUT SCHMIDT UNIVERSITÄT
Universität der Bundeswehr Hamburg

Angewandte Mathematik und Optimierung Schriftenreihe
Applied Mathematics and Optimization Series
AMOS # 17(2014)

Armin Fügenschuh, Konstanty Junosza-Szaniawski,
and Sławomir Kwasiborski

The Reservation-Allocation Network Flow Problem

Herausgegeben von der
Professur für Angewandte Mathematik
Professor Dr. rer. nat. Armin Fügenschuh

Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg
Fachbereich Maschinenbau
Holstenhofweg 85
D-22043 Hamburg

Telefon: +49 (0)40 6541 3540
Fax: +49 (0)40 6541 3672

e-mail: appliedmath@hsu-hh.de
URL: http://www.hsu-hh.de/am

Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Print 2199-1928
Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Internet 2199-1936

The Reservation-Allocation Network Flow Problem

Armin Fügenschuh[1], Konstanty Junosza-Szaniawski[2], and Sławomir Kwasiborski[2]

[1] Helmut Schmidt University / University of the Federal Armed Forces Hamburg, Holstenhofweg 85, 22043 Hamburg, Germany
fuegenschuh@hsu-hh.de
http://am.hsu-hh.de

[2] Politechnika Warszawska, Matematyki
ul. Koszykowa 75, 00-662 Warszawa
szaniaws,kwasiborskis@mini.pw.edu.pl
http://mini.pw.edu.pl

Abstract. We introduce and analyse a problem related to maximum flow problem in networks, called reservation-allocation-problem. It is motivated by a practical application in the natural gas transportation industry. In this two-stage problem, we seek for maximizing reservation values instead of flows. A reservation is feasible if it admits to transport any amount of flow not greater than the reserved values. We conjecture it is coNP-complete problem. Our main result is a decision algorithm determining if a given reservation in feasible in a given network.

Keywords: Network Flows, Reservation, Allocation, Maximum Flow, Minimum Cut.

1 Introduction

The maximum flow problem in networks is one of the most prominent combinatorial optimization problems. Informally stated, the problem asks for the maximum amount of flow that can be pushed through a network, entering the network at some sources and leaving at sinks, and respecting the network's capacity bounds. For its solution various theoretically and practically efficient algorithms are known [3, 1, 2, 5].

We introduce and analyze a related problem to this maximum flow problem, called reservation-allocation-problem, that is motivated by a practical application in the natural gas transportation industry. In this two-stage problem, we seek for maximizing reservation values instead of flows. A reservation is feasible if it admits to transport any amount of flow not greater than the reserved values (per node). A simple example shows that this problem can not be directly reduced to the max-flow problem. In order to check if a given reservation is feasible one has in principle to check an infinite number of possible flows. In this article we show that this is in fact not necessary. It is enough to restrict to a finite subset. The reservation-allocation is a coNP problem. It stays an open question if it is coNP-complete.

Our work is inspired by recent deregulation efforts in the natural gas market. Due to new European laws, the gas network operator is no longer identical with the gas owners. The operator has the only duty to transport the owners' gas from the entry nodes to their respective customers at exit nodes. For this the operator sells at each entry and exit nodes a reservation capacity (called "booking" in this industry). Gas owners (sellers) and gas consumers (buyers) then have the right to allocate any amount of gas up to the reserved capacity (allocation is called "nomination" in this industry). In reality one has to deal with further physical and technical aspects of gas transportation, for example, a nonlinear pressure loss of the gas during the transport, and technical equipment to increase or decrease the pressure (by compressors and control valves), and to take control of the flow by discrete routing decisions (opening and closing pipelines by valves). For more information on the general background of recent development of the gas industries, we refer to [4]. The nomination problem as a mathematical optimization problem is described in [9]. The booking problem is described in [4, 8], and also in the forthcoming book [7].

The remainder of this article is organized as follows. In Section 2 we introduce our notation while describing the classical network flow problem. In section 3 we present our main results, in section 4 we present some improvements of algorithms presented in section 3 and in section 5 we discuss open problems related to our work.

2 Preliminaries and problem formulation

A network is defined as a directed graph $G = (V^e, V^i, V^x, A, u)$, where V^e, V^i, V^x, A are finite sets, and $A \subseteq (V^e \cup V^i \cup V^x) \times (V^e \cup V^i \cup V^x)$ with $i \neq j$ for all $(i, j) \in A$. The node set (or vertex set) $V = V^e \cup V^i \cup V^x$ is partitioned into three pairwise disjoint subsets of entry nodes (or sources) V^e, exit nodes (or sinks) V^x and inner nodes V^i, where V^e and V^x are non-empty. Each arc (i, j) in the arc set A has a capacity $u_{i,j} \in \mathbb{R}_+$. A (feasible) flow in G is a vector $(a, f) \in \mathbb{R}_+^{V^e \cup V^x} \times \mathbb{R}_+^A$ that fulfils the following properties:

Capacity restriction on arcs:

$$f_{i,j} \leq u_{i,j}, \quad \forall (i, j) \in A. \tag{2.1}$$

Flow conservation in nodes:

$$\sum_{i:(i,j)\in A} f_{i,j} + a_j = \sum_{k:(j,k)\in A} f_{j,k}, \quad \forall j \in V^e, \tag{2.2a}$$

$$\sum_{i:(i,j)\in A} f_{i,j} = a_j + \sum_{k:(j,k)\in A} f_{j,k}, \quad \forall j \in V^x, \tag{2.2b}$$

$$\sum_{i:(i,j)\in A} f_{i,j} = \sum_{k:(j,k)\in A} f_{j,k}, \quad \forall\, j \in V^i. \tag{2.2c}$$

The components a_i for $i \in V^e$ are called the inflow into the network, and for $i \in V^x$ they are called the outflow. A necessary condition for a flow to be feasible is the balance of the total in- and outflow

$$\sum_{i\in V^e} a_i = \sum_{i\in V^x} a_i. \tag{2.3}$$

Because of this equality we can define the flow value for a feasible flow (a, f) as either one side of this equation, for example:

$$\text{flow}(G; a, f) := \sum_{i\in V^e} a_i. \tag{2.4}$$

Connected with the notion of flow is that of a cut. A cut C in G is a subset of arcs with the property that every path from any $i \in V^e$ to any $j \in V^x$ uses at least one arc in C. When removing all arcs in C from G, there is no more connection between entry and exit nodes. The capacity of a cut equals the sum of capacities of its arcs:

$$\text{cut}(G; C) := \sum_{(i,j)\in C} u_{i,j}. \tag{2.5}$$

Since any feasible flow (a, f) between sources and sinks must cross any cut C in G, we have that

$$\text{flow}(G; a, f) \le \text{cut}(G; C) \tag{2.6}$$

A famous result of Ford and Fulkerson states that for the maximum flow and the minimum cut equality holds:

$$\begin{aligned} \text{maxflow}(G) &:= \max\{\text{flow}(G; a, f) : (a, f) \text{ feasible flow in } G\} \\ &= \min\{\text{cut}(G; C) : C \text{ cut in } G\} =: \text{mincut}(G). \end{aligned} \tag{2.7}$$

Ford and Fulkerson also described an algorithm to find such maximum flow in polynomial time. In the following decades their algorithm was improved and also other methods for constructing maximal flows were described [6].

Let G be a network as defined above. An allocation is a vector $a \in \mathbb{R}_+^{V^e \times V^x}$. A balanced allocation is an allocation with

$$\sum_{i\in V^e} a_i = \sum_{i\in V^x} a_i. \tag{2.8}$$

An allocation is feasible if there exists a vector $f \in \mathbb{R}^A$ such that (a, f) is a feasible flow in the network. Note that a feasible allocation is necessarily balanced.

A reservation is a vector $r \in (\mathbb{R}_+ \cup \infty)^{V^e \times V^x}$. A reservation is feasible if any balanced allocation $a \in \mathbb{R}_+^{V^e \times V^x}$ with $0 \le a \le r$ is feasible (by $0 \le a \le r$ we mean $\forall v \in V^e \times V^x$ $0 \leq a_v \leq r_v$). Note that a reservation does not need to be balanced. We say that a is an allocation for r, if a is balanced and $0 \leq a \leq r$.

We define the following extension of the network $G = (V^e, V^i, V^x, A, u)$ and $S \subseteq V^e$ and $T \subseteq V^x$ in the following way. Let $G^r_{S,T} := (\{s\}, V^e, V^i, V^x, \{t\}, A_{S,T}, u_{S,T})$ where $A_{S,T} := A \cup \{(s,i) : i \in S\} \cup \{(i,t) : i \in T\}$ and $u_{S,T}(i,j) = u(i,j)$ for $(i,j) \in A)$, $u_{S,T}(s,i) = r(i)$ for $i \in V^e$, $u_{S,T}(i,t) = r(i)$ for $i \in V^x$. As abbreviation, we also write $G_{S,T}$ for $G^r_{S,T}$, if the dependency is clear from the context.

The following three problems are of interest.

Reservation Validation. Given: Network: $G = (V^e, V^i, V^x, A, u)$ and reservation $r \in (\mathbb{R}_+ \cup \infty)^{V^e \times V^x}$
Question: Is a given reservation r feasible. If it is not feasible, then a certificate, i.e., an infeasible allocation a for r, is to be determined. We can state two special cases of the reservation validation problem, motivated by real problems in gas networks: **Entry-bounded** (and exit-unbounded) - in this variant we can take any value on exits and have to impose the transport limitations of the network only on the entry side. The **Exit-bounded** (and entry-unbounded) case is defined in analogous way.

Subset Flow Problem. Given: Network: $G = (V^e, V^i, V^x, A, u)$ and reservation r,
Question: Weather for all sets $S \subseteq V^e, T \subseteq V^x$ hold
$\text{maxflow}(G^r_{S,T}) \geq \min(\sum_{v \in S} r_v, \sum_{v \in T} r_v)$?

Reservation Value Optimization. Given: Network: $G = (V^e, V^i, V^x, A, u)$ and a weight vector $w \in \mathbb{R}^{V^e \times V^x}$. This vector represents the value per unit entry or exit reservation capacity.,
Question: Find a feasible reservation vector r, such that $w^T r$ is maximal.

Let us consider a simple example of a network of a shape of a letter H, shown in Figure 2.1, left (cf. [7]). Let $V^e = \{s_L, s_R\}, V^i = \{i_L, i_R\}, V^x = \{x_L, x_R\}, A = \{s_L i_L, s_R i_R, i_L i_R, i_L x_L, i_R x_R\}$, $u(s_L i_L) = u(s_R i_R) = u(i_L x_L) = u(i_R x_R) = 10$ and $u(i_L i_R) = u(i_R i_L) = 1$. Notice that there exists a flow with a value 20: $f(s_L i_L) = f(s_R i_R) = f(i_L x_L) = f(i_R x_R) = 10$ (Figure 2.1, middle), but the reservation $r(s_L) = r(s_R) = 10$ is not valid. To show it is not valid it is enough to consider a allocation $a(s_L) = 10, a(s_R) = 0, a(x_L) = 0, a(x_R) = 10$ (Figure 2.1, right).

3 Main result

The Subset Flow Problem is obviously a subproblem of Reservation Validation problem. Our main result states that the Reservation Validation problem can be reduced to Subset Flow Problem, i.e., it is enough to verify only a finite set of allocations.

Theorem 1. *The Subset Flow Problem and Reservation Validation Problem are equivalent.*

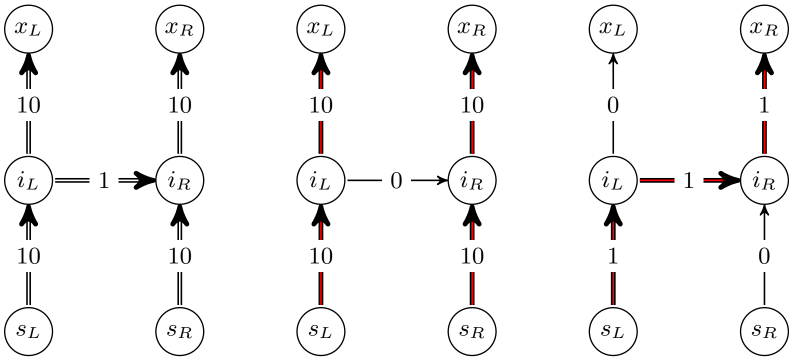

Fig. 2.1: Network example for a reservation capacity strictly less than maximum flow.

Proof. We show that an allocation $a \leq r$ with $\text{maxflow}(G^a_{V^e,V^x}) < \sum_{v \in V^e} a_v$ occurs, if and only if there exist sets that $S \subseteq V^e, T \subseteq V^x$, for which $\text{maxflow}(G^r_{S,T}) < \min\{\sum_{v\in S} r_v, \sum_{v \in T} r_v\}$ occurs.

1. We show that if sets S, T such that $\text{maxflow}(G^r_{S,T}) < \min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\}$ exist then r is infeasible. We assume that $F := \text{maxflow}(G^r_{S,T})$ and without loss of generality we assume that $\min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\} = \sum_{v\in S} r_v$. Therefore $F < \sum_{v\in S} r_v$.
   We define an allocation as follows:
$$a_i = \begin{cases} r_i, & i \in S, \\ r_i \dfrac{\sum_{v\in S} r_v}{\sum_{v\in T} r_v}, & i \in T, \\ 0, & i \in (V^e \cup V^x) \setminus (S \cup T). \end{cases}$$
   Clearly, $0 \leq a \leq r$. If there exists feasible flow for this allocation then $F = \sum_{v\in V^e} r_v = \sum_{v \in S} r_v$ which contradicts the assumption that $F < \sum_{v\in S} r_v$.

2. We will show that if there exists infeasible allocation a, then there exist sets S, T such that $\text{maxflow}(G^r_{S,T}) < \min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\}$. Let M_{V^e,V^x} be the minimal cut in graph $G^a_{V^e,V^x}$. From the fact that a is infeasible, we get that $\text{mincut}(G^a_{V^e,V^x}) = \text{maxflow}(G^a_{V^e,V^x}) < \sum_{v\in V^e} a_v$. Let $m_{V^e,V^x} := \text{mincut}(G^a_{V^e,V^x})$. We define sets S, T as follows:
$$S = \{v \in V^e \mid (s, v) \notin M_{V^e,V^x}\},$$
$$T = \{v \in V^x \mid (v, t) \notin M_{V^e,V^x}\}.$$
   We will show that $|S| \geq 1$. Let us assume, on the contrary, that S is empty. Then M_{V^e,V^x} contains all arcs that exit from s. Therefore, the sum of capacities of arcs belonging to M_{V^e,V^x} is equal to $m_{V^e,V^x} = \sum_{v\in V^e} a_v$, which

contradicts assumption that $m_{V^e,V^x} < \sum_{v \in V^e} a_v$. Hence $|S| \geq 1$. By the same arguments we show that $|T| \geq 1$.
Let $M_{S,T} := M_{V^e,V^x} \setminus (\{(s,i) \mid i \notin S\} \cup \{(i,t) \mid i \notin T\})$, then $M_{S,T}$ is a cut in the graph $G^r_{S,T}$. Its cut value (the sum of capacities of arcs that belong to $M_{S,T}$) is equal to $m_{V^e,V^x} - (\sum_{v \in V^e \setminus S} a_v + \sum_{v \in V^x \setminus T} a_v) = m_{S,T}$. Note that $\sum_{v \in V^e} a_v = \sum_{v \in V^x} a_v = \min\{\sum_{v \in V^e} a_v, \sum_{v \in V^x} a_v\}$. From the above, the cut $M_{S,T}$ is equal to:

$$
\begin{aligned}
m_{S,T} = m_{V^e,V^x} &- \sum_{v \in V^e \setminus S} a_v - \sum_{v \in V^x \setminus T} a_v \\
&< \min\{\sum_{v \in V^e} a_v, \sum_{v \in V^x} a_v\} - \sum_{v \in V^e \setminus S} a_v - \sum_{v \in V^x \setminus T} a_v \\
&= \min\{\sum_{v \in V^e \setminus S} a_v + \sum_{v \in S} a_v, \sum_{v \in V^x \setminus T} a_v + \sum_{v \in T} a_v\} \\
&\quad - \sum_{v \in V^e \setminus S} a_v - \sum_{v \in V^x \setminus T} a_v \\
&= \min\{\sum_{v \in V^e \setminus S} a_v + \sum_{v \in S} a_v - \sum_{v \in V^e \setminus S} a_v - \sum_{v \in V^x \setminus T} a_v, \\
&\quad \sum_{v \in V^x \setminus T} a_v + \sum_{v \in T} a_v - \sum_{v \in V^e \setminus S} a_v - \sum_{v \in V^x \setminus T} a_v\} \\
&= \min\{\sum_{v \in S} a_v - \sum_{v \in V^x \setminus T} a_v, \sum_{v \in T} a_v - \sum_{v \in V^e \setminus S} a_v\} \\
&\leq \min\{\sum_{v \in S} a_v, \sum_{v \in T} a_v\} \\
&\leq \min\{\sum_{v \in S} r_v, \sum_{v \in T} r_v\}.
\end{aligned}
$$

From the above, it follows that $\text{maxflow}(G^r_{S,T}) = \text{mincut}(G^r_{S,T}) \leq m_{S,T} < \min\{\sum_{v \in S} r_v, \sum_{v \in T} r_v\}$.

This Theorem gives us a natural algorithm for Reservation Validation Problem.

The algorithm either terminates with returning sets S, T and a deficit flow value F, which is less than the sum of reservation, or it terminates without a message, which means that the reservation is feasible.

The above algorithm requires to iterate over all subsets of a given set, hence it has an exponential running time. This raises the question if there might be a more efficient algorithm.

Corollary 1. *The output of the Algorithm I is correct.*

Algorithm 1 RESERVATION-VALIDATION(G, r)

for all $S \subseteq V^e$ **do**
    for all $T \subseteq V^x$ **do**
        if $F := \text{maxflow}(G_{S,T}) < \min\left\{\sum_{i\in S} r_i, \sum_{i\in T} r_i\right\}$ **then**
            return (S, T, F)
        end if
    end for
end for
return $\emptyset$

4 Improving the Algorithm

We speed up the algorithm by not taking all subsets S and T of V^e, V^x into consideration.

Lemma 1. *Let $S \subset V^e, T \subset V^x$. If $\min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\} = \sum_{v\in S} r_v$, meaning $\text{maxflow}(G^r_{S,T}) = \sum_{v\in S} r_v$, then for any T' with $T \subset T'$, we have that $\text{maxflow}(G^r_{S,T'}) = \sum_{v\in S} r_v$. Vice versa, if $\min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\} = \sum_{v\in T} r_v$, meaning $\text{maxflow}(G^r_{S,T}) = \sum_{v\in T} r_v$, then for any S' with $S \subset S'$, we have that $\text{maxflow}(G^r_{S,T'}) = \sum_{v\in T} r_v$.*

Proof. Let $T \subset T' \subseteq V^x$, $G^r_{S,T} = (V_{S,T}, A_{S,T}, u, r)$, $G^r_{S,T'} = (V_{S,T'}, A_{S,T'}, u, r)$. Let f be a maximum flow in $G^r_{S,T}$ with value $\sum_{v\in S} r_v$. Note that f is a feasible flow in $G^r_{S,T'}$. Let $M = \{(s,j) \in A_{S,T} : j \in S\}$, then M_{V^e,V^x} is a cut in $G^r_{S,T}$ with value $\sum_{v\in S} r_v$. Hence by the Ford-Fulkerson Maxflow-Mincut Theorem [3], M_{V^e,V^x} is a minimum cut. Since in $G^r_{S,T'}$ we only added arcs to T', M_{V^e,V^x} is still a cut in this extended graph. Again by the Maxflow-Mincut Theorem, f is a maximum flow in $G^r_{S,T'}$.

We remark that a termination of the RESERVATION-VALIDATION algorithm occurs if and only if sets S and T are found with $\text{maxflow}(G^r_{S,T}) < \min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\}$. Now if we compute $\text{maxflow}(G^r_{S,T})$ for some sets S and T and obtain $\text{maxflow}(G^r_{S,T}) = \min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\} = \sum_{v\in S} r_v$, then by Lemma 1 we can already exclude checking $\text{maxflow}(G^r_{S,T'})$ for any superset T' with $T \subset T'$.

We introduce the notion of a *relevant pair of sets* S, T for which we have to actually carry out the computation of $\text{maxflow}(G^r_{S,T})$. Such pair S, T is relevant, if $\min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\} = \sum_{v\in S} r_v$ and for every $T^* \subset T$ it holds that $\min\{\sum_{v\in S} r_v, \sum_{v\in T^*} r_v\} = \sum_{v\in T^*} r_v$ **or** if $\min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\} = \sum_{v\in T} r_v$ and for every $S^* \subset S$ it holds that $\min\{\sum_{v\in S^*} r_v, \sum_{v\in T} r_v\} = \sum_{v\in S^*} r_v$. We exploit this observation now algorithmically.

Theorem 2. *Algorithm 2 is equivalent to Algorithm 1, but checks only relevant pairs (S, T).*

Proof. We first show that all and only relevant pairs are checked (S, T). Let (S_0, T_0) be a relevant pair. W.l.o.g. we assume that $\min\{\sum_{v\in S_0} r_v, \sum_{v\in T_0} r_v\} =$

Algorithm 2 FAST-RESERVATION-VALIDATION(G, r)

1: $\mathcal{B} = \{(S,T) : S \subseteq V^e, T \subseteq V^x, |S| = |T| = 1\}$
2: **while** $\mathcal{B} \neq \emptyset$ **do**
3:   Let S, T such that $(S,T) \in \mathcal{B}$
4:   **if** $F := \text{maxflow}(G^r_{S,T}) < \min\{\sum_{v\in S} r_v, \sum_{v\in T} r_v\}$ **then**
5:     **return** (S, T, F)
6:   **end if**
7:   **if** $\min\left(\sum_{v\in S} r_v, \sum_{v\in T} r_v\right) = \sum_{v\in S} r_v$ **then**
8:     **for all** $v \in V^e \setminus S$ **do**
9:       $\mathcal{B} = \mathcal{B} \cup \{(S \cup \{v\}, T)\}$
10:     **end for**
11:   **else**
12:     **for all** $v \in V^x \setminus T$ **do**
13:       $\mathcal{B} = \mathcal{B} \cup \{(S, T \cup \{v\})\}$
14:     **end for**
15:   **end if**
16:   $\mathcal{B} = \mathcal{B} \setminus \{(S,T)\}$
17: **end while**
18: **return** $\emptyset$

$\sum_{v\in S_0} r_v$ (the other case follows from symmetry). Let $i \in S_0$ and $j \in T_0$. The pair $(\{i\}, \{j\})$ is added to $\mathcal{B}$ in Line 1. In Line 9 and Line 13 we always extend one of the sets S or T (selected as a pair (S,T) in Line 2) by some node v. Hence at some iteration we obtain sets (S^*, T_0) or (S_0, T^*), where $S^* \subset S_0$ and $T^* \subset T_0$. Let us assume that we obtained (S^*, T_0). Since (S_0, T_0) is a relevant pair, the check in Line 7 is "true". Hence further elements v are added to S^* only in Line 9. Finally, we will obtain (S_0, T_0). (The other case is symmetric.)

Since we add new elements in Line 9 and Line 13 only to those respective side that is responsible for the minimum, the algorithm never constructs a pair of sets (S,T) that is not relevant.

Theorem 3. *For any $G = (V^e, V^i, V^x, A, u)$ and for any reservation r there are at least $2^{\min(|V^e|,|V^x|)-1} - 1$ irrelevant pairs and there are networks with such number of pairs.*

Proof. Let (S_1, T_1) be a pair such that $|S_1| = |T_1| = 1$ let us assume that $\min\left(\sum_{v\in S_1} r_v, \sum_{v\in T_1} r_v\right) = \sum_{v\in S_1} r_v$. Every pair (S_1, T), such that $T_1 \subsetneq T$ is an irrelevant pair. Number of different pairs (S_1, T) is equal to $2^{|V^x|-1} - 1$. From the above it follows that in general case number of irrelevant pairs is not smaller than $2^{\min(|V^e|,|V^x|)-1} - 1$.
Now we will show that there exist networks with such number of pairs. Let G be a network in which $V^x = \{t\}$ and $\forall S \subseteq V^e \min\left(\sum_{v\in S} r_v, \sum_{v\in V^x} r_v\right) = \sum_{v\in S} r_v$. Network G contains no irrelevant pairs because all irrelevant pairs are of the form (S,T), where $t \subsetneq T$ but since V^x is one element set such T does not exist.

In the sequel we consider the case of entry-bounded, exit-unbounded, which is symmetric to the case of entry-unbounded, exit-bounded. We show that in this case it is sufficient to run Algorithm I for all $S \subseteq V^e$ and all $t \in V^x$ (instead of all subsets T of V^x).

Lemma 2. *The output of Algorithm I is the same if step (2) is replaced by "For all $t \in V^x$" if we consider the entry-bounded, exit-unbounded case.*

Proof. Assume that for subsets $S \subseteq V^e$ and $T \subseteq V^x$ we have $\text{maxflow}(G_{S,T}) < \min\left\{\sum_{i \in S} r_i, \sum_{i \in T} r_i\right\} = \sum_{i \in S} r_i$. For any $t \in T$ we then have the estimation $\text{maxflow}(G_{S,\{t\}}) \leq \text{maxflow}(G_{S,T})$, since the first problem is a restriction of the second one (in other words: fewer exits lead to less flow). So both algorithms would return the infeasibility of the reservations. On the other, if Algorithm I returns feasible, then checking only $T = \{t\}$ would also return feasible.

5 Conclusions and open problems

The subset flow problem is coNP-problem. For a given $S \subseteq V^e$ and $T \subseteq V^x$ we can verify in polynomial time if $\text{maxflow}(G^r_{S,T}) \geq \min(\sum_{v \in S} r_v, \sum_{v \in T} r_v)$. It remains an open question if it is an coNP-complete.

Another open question is an efficient algorithm for Reservation Value Optimization.

References

1. E. A. Dinic. Algorithm for solution of a problem of maximum flow in a network with power estimation. *Soviet Math. Doklady*, 11:1277 – 1280, 1970.
2. J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the ACM*, 19(2):248 – 264, 1972.
3. L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.
4. A. Fügenschuh, B. Geißler, R. Gollmer, C. Hayn, R. Henrion, B. Hiller, J. Humpola, T. Koch, T. Lehmann, A. Martin, R. Mirkov, A. Morsi, J. Rövekamp, L. Schewe, M. Schmidt, R. Schultz, R. Schwarz, J. Schweiger, C. Stangl, M. Steinbach, and B. Willert. Mathematical Optimization for Challenging Network Planning Problems in Unbundled Liberalized Gas Markets. *Energy Systems*, 2013.
5. A. V. Goldberg and R. E. Tarjan. A new approach to the maximum flow problem. In *Annual ACM Symposium on Theory of Computing, Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, 1986.
6. L. R. Ford Jr. and D. R. Fulkerson. *Flows in Networks.* Princeton University Press, 2010.
7. T. Koch, D. Bargmann, M. Ebbers, A. Fügenschuh, B. Geißler, N. Geißler, R. Gollmer, U. Gotzes, C. Hayn, H. Heitsch, R. Henrion, B. Hiller, J. Humpola, I. Joormann, V. Kühl, T. Lehmann, R. Lenz, H. Leövey, A. Martin, R. Mirkov, A. Möller, A. Morsi, D. Oucherif, A. Pelzer, M. Pfetsch, L. Schewe, W. Römisch, J. Rövekamp, M. Schmidt, R. Schultz, R. Schwarz, J. Schweiger, K. Spreckelsen, C. Stangl, M. Steinbach, A. Steinkamp, I. Wegner-Specht, and B. Willert. *Evaluating Gas Network Capacities.* Book in preparation, 2014.

8. A. Martin, B. Geißler, C. Hayn, J. Humpola, T. Koch, T. Lehmann, A. Morsi, M. Pfetsch, L. Schewe, M. Schmidt, R. Schultz, R. Schwarz, J. Schweiger, M. Steinbach, and B. Willert. Optimierung Technischer Kapazitäten in Gasnetzen. *VDI-Berichte: Optimierung in der Energiewirtschaft*, 2157:105–115, 2011.
9. M. Pfetsch, A. Fügenschuh, B. Geißler, N. Geißler, R. Gollmer, B. Hiller, J. Humpola, T. Koch, T. Lehmann, A. Martin, A. Morsi, J. Rövekamp, L. Schewe, M. Schmidt, R. Schultz, R. Schwarz, J. Schweiger, C. Stangl, M. Steinbach, S. Vigerske, and B. Willert. Validation of Nominations in Gas Network Optimization: Models, Methods, and Solutions. *Optimization Methods and Software*, 2013.