Improved optimization models for potential-driven network flow problems via ASTS orientations*†

Kai Helge Becker Benjamin Hiller

July 9, 2020

Abstract

The class of potential-driven network flow problems provides important models for a range of infrastructure networks. For real-world applications, they need to be combined with integer models for switching certain network elements, giving rise to hard-to-solve MINLPs. We observe that on large-scale real-world *meshed* networks the relaxations usually employed are rather weak due to cycles in the network. To address this situation, we introduce the concept of *ASTS (acyclic source-transshipment-sink) orientations*, a generalization of bipolar orientations, as a combinatorial relaxation of feasible solutions of potential-driven flow problems, study their structure, and show how they can be used to strengthen existing relaxations and thus provide improved optimization models. Our computational results indicate that ASTS orientations can be used to derive much stronger bounds on the flow variables than existing bound tightening methods. We also show that our proposed model extensions yield significant performance improvements for an existing state-of-the-art MILP model for large-scale gas networks.

1 Introduction

Network operators for utility and infrastructure networks face difficult planning and operational problems [BGS09, GHHV12, KHPS15, GMSS18]. Due to regulation and increasing cost pressure, but also due to the availability of powerful solvers, modern optimization methods are more and more applied to reduce cost and to improve the quality of planning and operation. The complexity of the considered optimization problems increases for several reasons, for instance because geographically bigger networks are considered or the detail level is increased. For instance, realistic network models for parts of the German gas network have more than 4000 nodes and almost 4500 arcs [SAB⁺17]. For networks of this size, providing globally optimal solutions or at least good bounds for assessing solution quality is still a big challenge [GMSS18].

A key submodel for infrastructure networks for fluids, e.g., water and gas [Rag13, DLWB15, GHHV12], is the so-called *potential-driven nonlinear network flow*

*This paper extends the work presented in the working papers [HB18, BH18].

†This preprint was first published as [BH19].

problem. This problem features a so-called *potential* π_u for each node $u \in V$, and the flow q_a of an arc $a \in A$ is related to the difference of the potential of its end nodes via an arc-specific *potential loss function* ϕ_a. Formally, the potential-driven network flow problem for a digraph $D = (V, A)$, supply vector q^{nom}, and bounds on the flow q_a and potential π_u is given by

$$\min \quad c^T(q, \pi, s) \tag{1a}$$

$$q_v^{\text{nom}} + \sum_{a=uv\in A} q_a - \sum_{a=vu\in A} q_a = 0 \quad \text{for all } v \in V, \tag{1b}$$

$$\pi_u - \pi_v = \phi_a(q_a) \quad \text{for all } a \in A, \tag{1c}$$

$$\underline{q}_a \le q_a \le \overline{q}_a \quad \text{for all } a \in A, \tag{1d}$$

$$\underline{\pi}_u \le \pi_u \le \overline{\pi}_u \quad \text{for all } u \in V. \tag{1e}$$

An important property of this model is that the flow is always directed from higher to lower potential, i.e., the potentials induce an acyclic orientation of the arcs. In other words: The network arising from a *feasible* flow by orienting each network arc in the direction of the flow over this arc contains no directed cycle.

This property is only implicit in the model for potential-driven network flows, i.e., it is not represented by an explicit constraint, but follows from the interplay of flow conservation and potential loss along an arc. Therefore, it is not reflected in the relaxations used to solve these nonlinear noncovex optimization problems. This is discussed in more detail in [HB18].

Our contribution This paper proposes to use the discrete structure arising from the combination of flow conservation and an acyclicity requirement for the flow to strengthen MINLP models and their relaxations that arise from potential-driven network flow problems. These strengthened models are particularly valuable when using global optimization methods on large networks.

Based on our earlier work in [HB18, BH18], we introduce a combinatorial abstraction of feasible flows we call *ASTS (acyclic source-transshipment-sink) orientation.* This combinatorial structure is implicit in otherwise purely continuous nonconvex models. Hence our approach enables to exploit the power of discrete global optimization techniques in this context. As real networks contain additional network elements that do not necessarily fit the potential-driven network flow framework, we consider a generic MINLP featuring the potential-driven network flow problem as a submodel. Moreover, we consider not only fixed demand vectors, but intervals for the demand at each source/sink. This is useful for situations where the demand is not fully fixed, e.g., due to mixing of gas qualities [GMSS15]. Moreover, this may be applied to analyze the difference in flow directions if large classes of demand vectors are possible [HHH$^+$15].

We provide an analysis of the structure of ASTS orientations, aiming at a generic theory and large-scale networkks. In [HB18] we observed that in order to have a combinatorial model capturing both acyclicity and flow conservation, it is necessary to consider three "directions" for each arc of the network: forward flow, backward flow, and zero flow, yielding the concept of *acyclic flow orientations.* In contrast, ASTS orientations admit only the two classical directions forward and backward. A careful analysis of the structure of ASTS orientations (Theorem 1) allows us to identify maximal subgraphs for which an

ASTS orientation exists. We prove that ASTS orientations provide a combinatorial relaxation for the MINLP model mentioned above (Theorem 2), showing that arcs outside the maximal subgraph admitting an ASTS orientation carry zero flow. Hence we are able to deal with two possible directions only, which is crucial to reduce both model and computational complexity. We study decompositions of graphs admitting ASTS orientations and how they can be used to describe the sets of the corresponding ASTS orientations (Theorem 3, Theorem 4, Theorem 5). Based on these decompositions, we outline a possible algorithmic scheme for exploiting the structure of ASTS orientations computationally. Our computational results indicate that this is very useful to tackle large-scale real-world networks.

Related work Potential-driven network flow problems have been considered at least since the 1970s [Mau77, CCH$^+$78] and are relevant for a variety of applications areas [HJ84]. Although *pure* potential-driven network flow problems (1) are nonconvex NLPs, their special structure enables the development of efficient solution techniques [CCH$^+$78, VMC15]. For network design problems, binary variables for enabling/disabling network elements are introduced, yielding MINLPs that are NP-hard [GPS$^+$19] and significantly harder to solve in practice [Rag13, HF13]. The same holds when the problem is extended by models for further (switchable) network elements to model network operation in detail [PFG$^+$15, KHPS15, GMSS18].

For the case of the potential-driven network flow problem (1) with potential loss function $\phi_a(q_a) = c_a q_a |q_a|$, knowing the flow directions for all arcs enables the use of algebraic methods to e.g., solve certain stochastic optimization variants of the problem [GHHS16, GNS17]. Moreover, in many relevant cases the function describing the potential difference between two nodes is convex and fixed flow directions result in convex MINLP problems [Rag13]. Exploiting the structural results presented here in enumerating ASTS orientations [BH18] will support these lines of research.

To the best of our knowledge, there are only two other papers that exploit the acyclicity property of potential-driven flows computationally. In [HPU19], the authors extend their model by dicycle inequalities [GJR85] to ensure acyclicity of the flows. However, they do not use a combinatorial model for flow conservation. The very recent preprint [HP20] details the method from [HPU19] and expands it to incorporate a combinatorial model for flow conservation as in [HB18]. Starting from the polytope of orientations corresponding to potential-driven flows, [HP20] considers several relaxations, arriving at a polytope $\mathcal{P}^{\mathrm{AS}\pm}$ that is essentially the polytope of acyclic flow orientations. In contrast to this paper and our earlier work in [HB18], orientations are only considered for the potential-driven arcs. Moreover, two additional binary variables per arc are used to encode the three possible flow directions, while we need only one binary variable per arc, yielding a more compact and also tighter formulation. In their analysis of $\mathcal{P}^{\mathrm{AS}\pm}$, the authors rediscover several properties of acyclic flow/ASTS orientations that have been established in [HB18] and [BH18] and are also presented in this paper. Our proposed modeling approach of enumerating ASTS orientations and adding set-partitioning constraints for selecting an orientation can be seen as a Dantzig-Wolfe reformulation of the variant of $\mathcal{P}^{\mathrm{AS}\pm}$ which uses only one binary variable for the flow direction. In combination with our decomposition results this enables us to tackle larger instances. Still, the computational results in [HP20] demonstrate the strong benefit from using a combinatorial relaxation of potential-driven flows and hence support our approach based on ASTS orientations.

2 General model for potential-driven network flow problems

This section introduces the general model for potential-driven network flow problems that this paper is concerned with and provides some basic notation.

Let $G = (V, E)$ be a graph that represents the network the flow on which we seek to optimize. For the purpose of formulating our MINLP model we define $D = (V, A)$ to be the digraph arising by replacing each edge $e = \{u, v\}$ by a forward arc uv and a backward arc vu, i.e., $A = \{uv, vu \mid \{u, v\} \in E\}$. (We note that the theory developed in this paper also applies to multigraphs and multidigraphs, i.e., for networks with parallel arcs, but for ease of notation and language we will speak of graphs and digraphs in the following.) We denote the arc set induced by an edge set $E' \subseteq E$ by $A[E']$. Analogously, we define the set $E[A']$ to be the edges corresponding to a set of arcs A'. To model network elements occuring in applications, we assume the edge set E is a partition of sets of potential-decreasing edges E_{dec}, potential-maintaining edges E_{mnt}, and generic edges E_{gen}. The operation of the potential-increasing and generic edges is described by feasible sets $F_e^{\text{dec}}, F_e^{\text{gen}} \subseteq \mathbb{R}^4_{\geq 0}$, that relate the feasible values of the inlet potentials π_u, the outlet potentials π_v and the flows q_{uv}, q_{vu} on the arcs uv, vu for $e = \{u, v\} \in E_{\text{dec}}$. The feasible sets F_e^{dec} for $e = \{u, v\} \in E_{\text{dec}}$ are characterised by

$$F_e^{\text{dec}} \subseteq \{(\pi_u, \pi_v, q_{uv}, q_{vu}) \in \mathbb{R}^4_{\geq 0} \mid q_{uv} - q_{vu} > 0 \implies \pi_u > \pi_v,\\ q_{uv} - q_{vu} < 0 \implies \pi_u < \pi_v\}. \tag{2}$$

For generic edges, there are no assumptions about the feasible set F_e^{gen}. Hence they cover all network elements not fitting our framework of potential-decreasing and -maintaining elements, e.g., compressors in gas networks or pumps in water networks. Potential-maintaining edges may be closed or open valves. In the open state the potential at their ends is equal, whereas in the closed state the flow is zero.

In practical applications the vast majority of edges is typically in E_{dec} and represents pipelines that can be modelled by a non-linear potential-loss function $\phi_e \colon \mathbb{R} \to \mathbb{R}$ based on the edge flow $q_e := q_{uv} - q_{vu}$ for edge $e = \{u, v\}$ with

$$\begin{aligned} \phi_e(q_e) > 0 &\iff q_e > 0, \\ \phi_e(q_e) < 0 &\iff q_e < 0, \\ \phi_e(q_e) = 0 &\iff q_e = 0, \end{aligned}$$

in which case the feasible set is given by

$$F_e^{\text{dec}} = \{(\pi_u, \pi_v, q_{uv}, q_{vu}) \in \mathbb{R}^4_{\geq 0} \mid \pi_u - \pi_v = \phi_e(q_{uv} - q_{vu})\}.$$

The variables in our model are given by the flows $q_a \in \mathbb{R}_{\geq 0}$ on the arcs $a \in A$, the flows $q_u \in \mathbb{R}$ into and out of the nodes $u \in V$, representing sources and sinks, the potentials $\pi_u \in \mathbb{R}_{\geq 0}$ at the nodes $u \in V$, and variables $s_e \in \{0,1\}$ that indicate whether a potential-maintaining edge $e \in E_{\text{mnt}}$ is open or closed. We also include upper and lower bounds for flows on arcs, flows into and out of nodes and the potentials at nodes, namely $\overline{q}_a$, $\underline{q}_a$, $\overline{q}_u$, $\underline{q}_u$, $\overline{\pi}_u$, and $\underline{\pi}_u$, respectively. This means in particular that we allow for nodes with flexible demands (i.e., nodes are not necessarily predetermined as sources or sinks with a given in- or out-flow) and develop our theoretical framework in the following sections accordingly.

The (linear) costs in our MINLP are represented by a cost vector $c \in \mathbb{R}^{|V|+|A|+|V|+|E_{\text{mnt}}|}$ that models the costs for flows in and out of a node, costs for flows along arcs, costs related to the potential at each node, and costs for opening or closing an edge in E_{mnt}.

When required we will call a cycle $(u_0, u_1, \ldots, u_n, u_0)$, $u_0, u_1, \ldots, u_n \in V$, a negative-cost cycle if $\sum_{i=0}^{n-1} c_{u_i u_{i+1}} + c_{u_n u_0} < 0$, $\overline{q}_{u_i u_{i+1}} > 0$ for all $i = 0, 1, \ldots, n-1$, and $\overline{q}_{u_n u_0} > 0$. As each edge of the network is modelled by a forward and a backward arc in our MINLP, there may be solutions in which the model sends a flow through an edge in both directions. To exclude this unrealistic case for optimal solutions, we make the assumption throughout the paper that the cost vector is such that there are no negative 2-cycles, i.e. for all edges $e \in E$ if $c_{uv} + c_{vu} < 0$ then $\overline{q}_{uv} = 0$ or $\overline{q}_{vu} = 0$.

On this basis, our MINLP looks as follows:

$$\min c^T(q, \pi, s) \tag{3a}$$

$$\text{s. t. } q_v + \sum_{a=uv \in A} q_a - \sum_{a=vu \in A} q_a = 0 \quad \text{for all } v \in V, \tag{3b}$$

$$(\pi_u, \pi_v, q_{uv}, q_{vu}) \in F_e^{\text{dec}} \quad \text{for all } e = \{u,v\} \in E_{\text{dec}}, \tag{3c}$$

$$(\pi_u, \pi_v, q_{uv}, q_{vu}) \in F_e^{\text{gen}} \quad \text{for all } e = \{u,v\} \in E_{\text{gen}}, \tag{3d}$$

$$s_e = 0 \implies q_{uv} = q_{vu} = 0 \quad \text{for all } e = \{u,v\} \in E_{\text{mnt}}, \tag{3e}$$

$$s_e = 1 \implies \pi_u = \pi_v \quad \text{for all } e = \{u,v\} \in E_{\text{mnt}}, \tag{3f}$$

$$s_e \in \{0,1\} \quad \text{for all } e \in E_{\text{mnt}}, \tag{3g}$$

$$\underline{q}_v \leq q_v \leq \overline{q}_v \quad \text{for all } v \in V, \tag{3h}$$

$$0 \leq \underline{q}_a \leq q_a \leq \overline{q}_a \quad \text{for all } a \in A, \tag{3i}$$

$$0 \leq \underline{\pi}_v \leq \pi_v \leq \overline{\pi}_v \quad \text{for all } v \in V. \tag{3j}$$

The flow part q of a MINLP solution (q, π, s) naturally induces a digraph $D(E', q) := (V(q), A(q))$ for any edge subset $E' \subseteq E$ of the underlying graph G via

$$\begin{aligned} A(q) &:= \{a \in A[E'] \mid q_a > 0\}, \\ V(q) &:= V[A(q)]. \end{aligned} \tag{4}$$

We say that the solution is *cyclic* if $D(E', q)$ contains a cycle and *acyclic* otherwise. Observe that an arc with $\overline{q}_a = 0$ is never in a cycle.

3 ASTS-orientations and their basic properties

We now define the theoretical concept that is central to this paper. It captures the discrete structure that arises from the combination of the flow conservation constraint and the acyclicity that follows from the fact that flow is always directed from higher to lower potential. Recall that an orientation of an underlying graph $G = (V, E)$ is a digraph $\tilde{D} = (V, \tilde{A})$ with $uv \in \tilde{A} \iff vu \notin \tilde{A}$ for all $\{u, v\} \in E$.

Definition 1 Let $\tilde{D} = (V, \tilde{A})$ be a digraph with an underlying graph G, with $V_+, V_-, V_T \subsetneq V$ being disjoint sets of "sources", "sinks" and "transshipment nodes", respectively. We call the nodes in $V_F := V - V_+ - V_- - V_T$ "free nodes". A node $u \in V$ is said to satisfy the source-transshipment-sink-condition (STS-condition) with respect to (V_+, V_-, V_T, V_F) if

(i) $u \in V_+$ and there exists an arc $a = uv \in \tilde{A}$, or

(ii) $u \in V_-$ and there exists an arc $a = vu \in \tilde{A}$, or

(iii) $u \in V_T$ and there exist arcs $a_1 = vu, a_2 = uw \in \tilde{A}$, or

(iv) $u \in V_F$.

If all nodes $u \in V$ satisfy the STS-condition and $\tilde{D}$ is acyclic, $\tilde{D}$ is called an ASTS-orientation of G with respect to (V_+, V_-, V_T, V_F), and G is said to have an ASTS-orientation.

A subgraph $G_0 = (V_0, E_0)$ of G is said to have an ASTS-orientation with respect to (V_+, V_-, V_T, V_F) if there exists an orientation $\tilde{D}_0$ of G_0 such that $\tilde{D}_0 = (V_0, \tilde{A}_0)$ is an ASTS-orientation of G_0 with respect to $(V_+ \cap V_0, V_- \cap V_0, V_T \cap V_0, V_F \cap V_0)$.

Moreover, we define the sets of *in-nodes* $V_{+,F} := V_+ \cup V_F$ and *out-nodes* $V_{-,F} := V_- \cup V_F$. □

Remark 1 Observe that the common property of in-nodes is that both sources and free nodes do not require an in-arc, while analogously out-nodes do not require an out-arc. □

Note that ASTS-orientations are a generalization of bipolar orientations (see e.g., [dFOdMR95] for the concept of bipolar orientations). Moreover, there exists a bijection between the set of all ASTS-orientations of a graph G with $V_F = \varnothing$ and the bipolar orientations of the extended graph that arises from adding to G a "supersource" s that is adjacent to all sources, a "supersink" t that is adjacent to all sinks, and an edge $\{s, t\}$.

The following theorem characterizes graphs that allow for an ASTS-orientation.

Theorem 1 *Let $G = (V, E)$ be a connected graph with node sets V_+, V_-, V_T, V_F as in Definition 1. Then the following four statements are equivalent:*

(i) G has an ASTS-orientation $\tilde{D} = (V, \tilde{A})$ with respect to (V_+, V_-, V_T, V_F).

(ii) (path characterization) *Every node $u \in V$ is on a path from an in-node to a distinct out-node.*

(iii) (block characterization) *G contains at least one in-node and a distinct out-node, and the leaves of the block-cut tree of G correspond to blocks that have an in- or out-node distinct from the cut-node of the block.*

(iv) (completion characterization) *There exists a subgraph of G that has an ASTS-orientation, and for every ASTS-orientation $\tilde{D}_0 = (V_0, \tilde{A}_0)$ of some subgraph $G_0 = (V_0, E_0)$ of G there exists an orientation $\tilde{A}_1$ of the edges $E \setminus E_0$ such that $\tilde{D} = (V, \tilde{A}_0 \cup \tilde{A}_1)$ is an ASTS-orientation of G with respect to (V_+, V_-, V_T, V_F).* □

PROOF We will prove the theorem in the following order:

$$(i) \implies (ii) \implies (iii) \implies (ii) \implies (iv) \implies (i).$$

$(i) \implies (ii)$ Let $\tilde{D} = (V, \tilde{A})$ be an ASTS-orientation of G with respect to (V_+, V_-, V_T, V_F). We choose a node $u \in V$. If u is an in-node, we follow a directed path via out-arcs, starting from u. As $\tilde{D}$ is acyclic we will not return to a node we have already visited, and because G is finite, the directed path will end at a node that has no out-arcs, i.e., an out-node. Similarly, if u is an out-node, we can follow a directed path from u via in-arcs until we find an in-node. If u is a transshipment node, we can construct one directed path to an in-nod and another directed path to an out-node, where the acyclicity of $\tilde{D}$ ensures that the two paths are node-disjoint. Since we can carry out this procedure for any node, all nodes are on a directed path from an in-node to a distinct out-node. Now (ii) follows from considering the graph underlying $\tilde{D}$.

$(ii) \implies (iii)$ We show the contrapositive. Clearly, if G does not contain an in-node and a distinct out-node, condition (ii) cannot be satisfied. Moreover, if a block of a graph is a leaf of the block-cut tree and all nodes other than the cut-node of the block are transshipment nodes, a path from any of these nodes to an in- or out-node must contain the cut-node. Hence, no node of the block, except possibly the cut-node itself, is on a path from an in-node to an out-node.

$(iii) \implies (ii)$ Let $u \in V$ be a node of G. Now if u is an in-node (out-node) itself, it is clearly on a path from an in-node to an out-node as it is connected with the distinct out-node (in-node) of G. Therefore, in the following let u be a transshipment node. We distinguish between three cases.

Case 1: The node u is in a block of G that has both an in-node and an out-node. By virtue of the 2-connectedness of the block there exist two internally disjoint paths from u to an in-node and two internally disjoint paths from u to an out-node. As a consequence, u must be on a path from an in-node to an out-node.

Case 2: The node u is in a block that has either an in-node or an out-node. We assume w.l.o.g. that the block has an in-node. If the block is a leaf of the block-cut tree of G, the in-node is not the cut-node of the block due to condition (iii). If the block is not a leaf of the block-cut tree, we will assume for the moment that the in-node is not a cut-node of the block. Now, since G has an in-node and a distinct out-node, the block must have a cut-node that is connected with an out-node in a different block. Then, due to the 2-connectedness of blocks, there exist two internally disjoint paths from u to the in-node and from u to the cut-node, with the latter being the case only if u is not the cut-node itself. Hence there is a path from the in-node via u via the cut-node to an out-node.

Now let us consider the case where our block is not a leaf of the block-cut tree and the in-node is a cut-node of our block. Then the block contains at least two cut-nodes and one of the other cut-nodes of the block must be providing us with a path to an in-node or an out-node in a further block, otherwise the block-cut tree of G would have a leaf without in- and out-nodes. If the cut-nodes provide us with a path to an out-node, there clearly exists, again due to the 2-connectedness of blocks, a path from this block's in-node to an out-node that passes through u. If the cut-nodes only provide us with a path to another in-node, there must be another out-node in a block that is accessible via the cut-node that is an in-node, otherwise G would not have an out-node. As a consequence, u is again on a path from an in-node to an out-node.

Case 3: The node u is in a block without in-node and out-node. Then u cannot be in a block that is a leaf of the block-cut tree and therefore the block in which u is located contains at least two cut-nodes (one of which may be u itself). All cut-nodes must be on a path from u to an in-node or from u to an out-node, otherwise the block-cut tree of G would have a leaf without in- and out-nodes. Moreover, due to the fact that G has both an in-node and an out-node, one of the cut-nodes must be on a path from u to an in-node and a different cut-node must be on a path from u to an out-node. As a consequence, since a block is 2-connected, u must be on a path from an in-node to an out-node.

$(ii) \implies (iv)$ By assumption every node is on a path from an in-node to a distinct out-node. Clearly, by orienting one such path from the in-node to the out-node, we obtain an ASTS-orientation of a subgraph of G. Now let $\tilde{D}_0 = (V_0, \tilde{A}_0)$ be any ASTS-orientation of some proper subgraph of G. We will procede in two steps.

In the first step we will extend $\tilde{D}_0$ to an ASTS-oriented digraph whose underlying graph is still G and that contains all in- and out-nodes, provided the latter is not the case yet. In the second step we will show that, provided the graph underlying our ASTS-oriented digraph contains all in- and out-nodes and is a proper subgraph of G, we can always find a path on the remaining unoriented edge set of G that can be oriented to provide a larger ASTS-oriented digraph. Statement (iv) then follows by induction on the remaining unoriented edge set.

Step 1 We enlarge the digraph $\tilde{D}_0$ by connecting, with the out-nodes, all in-nodes that are not yet in the node set of our digraph, using directed paths whose internal nodes are not in the node set of our digraph either. The prodecure is as follows: for each in-node, we take an arbitrary path on G from the in-node to an out-node, which is possible due to (ii), and orient, into the direction away from the in-node, either the subpath from the in-node to the first node of the existing digraph or, if this is not possible because none of the nodes of the path is on the digraph, the entire path from the in-node to an out-node. In a similar fashion we add all remaining out-nodes to our digraph: by orienting edges that connect these out-nodes to the closest node of our existing digraph or by directly connecting them with an in-node. The resulting enlarged digraph $\tilde{D}_1 = (V_1, \tilde{A}_1)$ has an underlying graph that is a subgraph of G, has a node set that includes all in- and out-nodes, and is ASTS-oriented (note that we did not create any directed cycle because we did not connect any two nodes that were already in the node set of our digraph).

Step 2 We will now show that given an ASTS-oriented digraph $\tilde{D}_1$ with an underlying proper subgraph G_1 of G and with a node set that includes all in- and out-nodes, we can always find a path P with edges from $E - E_1$, with distinct endnodes in V_1, and with all other nodes being in $V - V_1$ that can be oriented such that the digraph that arises from adding the oriented path to $\tilde{D}_1$ again yields an ASTS-oriented digraph.

We pick an arbitrary edge from $E - E_1$. If both endnodes of this edge are in V_1, this edge is our path P. Otherwise, we extend this edge to a path by adding edges from $E - E_1$ until we have a path P with two distinct endnodes both of which are in V_1. This is always possible because of (ii) and since all in- and out-nodes are in V_1 according to the construction in Step 1. We denote the endnodes of P by $u, v \in V_1$ and the inner nodes of P by $V' \subseteq V \setminus V_1$.

We now orient P as a directed path, i.e. from one endnode to the other such that each internal node of the path is both head and tail of an arc, and denote the arcs of the directed path by $\tilde{A}'$. For constructing our orientation we observe that, as $\tilde{D}_1$ is acyclic and u and v are nodes of $\tilde{D}_1$, there are three cases:

(a) no directed path on $\tilde{D}_1$ from u to v and no directed path from v to u, or

(b) there is a directed path from u to v, but not from v to u, or

(c) there is a directed path from v to u, but not from u to v.

In case (a) we choose an arbitrary orientation for our directed path. In cases (b) and (c) we orient the path such that it has the same orientation as the existing path. As a consequence the orientation $\tilde{D}' = (V_1 \cup V', \tilde{A}_1 \cup \tilde{A}')$ that results from adding the arcs and nodes of the directed path to $\tilde{D}_1$ will be acyclic, too. (Note that by construction of P, only the endnodes of P are in V_1, and therefore orienting P cannot create any cycle containing an internal node of P.) As the STS-condition is already satisfied at the endnodes $u, v \in V_1$ and we have oriented the path P to form a directed path, the inner nodes of the path and hence the digraph $\tilde{D}'$ altogether also satisfy the STS-condition. Hence the resulting digraph $\tilde{D}' = (V_1 \cup V', \tilde{A}_1 \cup \tilde{A}')$ is ASTS-oriented.

$(iv) \implies (i)$ This is trivially the case. ■

We have seen in the previous theorem that not all graphs admit an ASTS-orientation. The block characterization of Theorem 1 implies that even if a graph does not have an ASTS-orientation, this may be the case for a subgraph. This raises the question of whether we can find a maximal subgraph of a given graph G that allows for an ASTS-orientation. Indeed, successively removing blocks from G will lead to such a maximal subgraph.

Definition 2 Let $G = (V, E)$ be a connected graph with node subsets (V_+, V_-, V_T, V_F). Consider the block-cut tree T of G and repeatedly execute the following steps for leaf nodes of the tree if possible:

1. Remove a leaf corresponding to a block if the block does not have a source, sink, or free node distinct from the cut-node of the block.

2. Remove a leaf corresponding to a cut-node if it is not a source, sink, or free node.

Let the *outer node set* V_{out} be the nodes of G corresponding to the nodes removed from T, and the *inner node set* V_{in} the set of nodes still represented by T.

In the special case that there is not an in-node and a distinct out-node, i.e.,

$$|V_{+,F}| = 0 \quad \text{or} \quad |V_{-,F}| = 0 \quad \text{or} \quad |V_{+,F} \cup V_{-,F}| \leq 1 \tag{5}$$

we instead set $V_{\text{out}} = V$ and $V_{\text{in}} = \emptyset$.

We refer to the edge set induced by V_{out} as *outer edges* $E_{\text{out}} \subseteq E$ and to the edge set induced by V_{in} as *inner edges* $E_{\text{in}} \subseteq E$. Further let $A_{\text{in}} := A[E_{\text{in}}] \subseteq A$ and $A_{\text{out}} := A[E_{\text{out}}] \subseteq A$ be the arc sets corresponding to the inner and outer edges, respectively. □

This definition is justified by the following insight.

Corollary 1 *Let $G = (V, E)$ be a graph with node subsets (V_+, V_-, V_T, V_F) and V_{in} and V_{out} the inner and outer node sets of G, respectively.*

(i) There is no subgraph of G that contains a node from $V_{\text{out}} \setminus V_{\text{in}}$ and has an ASTS-orientation with respect to (V_+, V_-, V_T, V_F).

(ii) The subgraph of G induced by V_{in} is the maximal subgraph of G to have an ASTS-orientation with respect to (V_+, V_-, V_T, V_F). □

Proof As no node in V_{out} is, by construction, on a path from an in-node to a distinct out-node statement (i) follows from the path characterization of Theorem 1. Statement (ii) follows from (i) by taking into account that the subgraph induced by V_{in} has an ASTS-orientation according to the block characterization of Theorem 1. ■

4 ASTS orientations as combinatorial relaxations for potential-driven network flow problems

In the following we will consider the relationship between ASTS-orientations and the solutions of MINLP (3). We have seen that in general only a subgraph of a given graph admits an ASTS-orientation. In particular, Corollary 1 characterizes the maximal subgraph with this property. We will see shortly that ASTS-orientations are a suitable combinatorial model for flows in potential flow problems. In this context, the interpretation for the subgraph $(V_{\text{out}}, A_{\text{out}})$ obstructing an ASTS-orientation is that the flows in this subgraph have to be zero. The reason is that this is the only way to comply with flow conservation and the fact that for potential-decreasing edges there cannot be positive flow cycles. We can extend this reasoning to potential-maintaining edges, assuming that the zero flow is "as good as" a positive flow cycle, i.e., both feasible and optimal. This is frequently the case in practical applications. Handling both potential-decreasing and potential-maintaining edges allows to tackle larger subnetworks with the techniques proposed in this paper. The technical details for these subnetworks in the case of MINLP (3) are given by the following definition.

Definition 3 An arc subset $\hat{A}$ of the digraph $D = (V, A)$ for MINLP (3) is called *ASTS-representable*, if it fulfills the following conditions:

(i) $\hat{A}$ does not contain generic edges, i.e., $\hat{A}[E_{\text{gen}}] = \emptyset$.

(ii) The flow for potential-maintaining edges can be zero, i.e., $\underline{q}_a = 0$ for all $a \in \hat{A}[E_{\text{mnt}}]$.

(iii) There is no negative-cost cycle on $\hat{A}[E_{\text{mnt}}]$. □

Definition 4 For a digraph $D = (V, A)$ for MINLP (3), an arc subset $A' \subseteq A$ is called a region. The node set $V' \subseteq V$ of the region is the node set induced by A', i.e. $V' := V[A']$, and the sources, sinks, transshipment nodes and free nodes of a region are given by:

$$\underline{q}_v^{A'} := \underline{q}_v + \sum_{a=uv \in A \setminus A'} \underline{q}_a - \sum_{a=vu \in A \setminus A'} \overline{q}_a, \tag{6a}$$

$$\overline{q}_v^{A'} := \overline{q}_v + \sum_{a=uv \in A \setminus A'} \overline{q}_a - \sum_{a=vu \in A \setminus A'} \underline{q}_a, \tag{6b}$$

$$V'_+ := \{v \in V' \mid \underline{q}_v^{A'} > 0\}, \tag{6c}$$

$$V'_- := \{v \in V' \mid \overline{q}_v^{A'} < 0\}, \tag{6d}$$

$$V'_T := \{v \in V' \mid \underline{q}_v^{A'} = \overline{q}_v^{A'} = 0\} \text{ and} \tag{6e}$$

$$V'_F := V' - V'_+ - V'_- - V'_T. \tag{6f}$$

□

The following lemma justifies our definition of sources, sinks and transshipment nodes of a region by showing that these function as sources, sinks and transshipment nodes for a region when considering the solutions of MINLP (3).

Lemma 1 *Let $G = (V, E)$ be a digraph for MINLP (3) and A' a region with sources, sinks, transshipment nodes and free nodes V'_+, V'_-, V'_T and V'_F, respectively.*

Then in any feasible solution (q, π, s) to MINLP (3)

(i) for all $u \in V'_+$ there exists an arc $a = uv \in A'$ with $q_a > 0$ for some $v \in V'$,

(ii) for all $u \in V'_-$ there exists an arc $a = vu \in A'$ with $q_a > 0$ for some $v \in V'$, and

(iii) for all $u \in V'_T$ there exists an arc $a_1 = vu \in A'$ with $q_{a_1} > 0$ for some $v \in V'$ iff there exists an arc $a_2 = uw \in A'$ with $q_{a_2} > 0$ for some $w \in V'$. □

Proof Let u be in V'_+. With (6c), (6a) and the node demand bounds (3h) and the flow bounds (3i), we obtain

$$\begin{aligned} 0 < \underline{q}_u^{A'} &= \underline{q}_u + \sum_{a=uv \in A \setminus A'} \underline{q}_a - \sum_{a=vu \in A \setminus A'} \overline{q}_a \\ &\le q_u + \sum_{a=uv \in A \setminus A'} q_a - \sum_{a=vu \in A \setminus A'} q_a, \end{aligned}$$

which, due to the flow conservation (3b) is equal to

$$= -\sum_{a=uv\in A'} q_a + \sum_{a=vu\in A'} q_a.$$

Now statement (i) follows with $q_a \geq 0$ for all $a \in A$. Statement (ii) follows analogously. For $u \in V'_T$ we proceed from (6e) in the same fashion as above, but once with (6a) and once with (6b). Using the node demand bounds (3h), the flow bounds (3i) and the flow conservation (3b) we obtain two inequalities that imply $\sum_{a=vu\in A'} q_a = \sum_{a=uv\in A'} q_a$. Then statement (iii) again follows from $q_a \geq 0$ for all $a \in A$. ■

We can now relate the solutions of MINLP (3) to the ASTS-orientations on regions of the arc set. The following theorem shows that under suitable conditions there exist ASTS-orientations that are compatible with the flow on regions. This result provides the reason why studying ASTS-orientations is useful when considering the potential-based network flow problem of MINLP (3).

Theorem 2 *Consider a subdigraph $D' = (V', A')$ of $D = (V, A)$ for MINLP (3) where A' is an ASTS-representable region and denote by A'_{in}, A'_{out}, V'_{in}, V'_{out} the inner and outer arc and node sets arising when applying Definition 2 to the underlying graph of D'. Further, consider the sets of sources V'_+, sinks V'_-, transshipment nodes V'_T and free nodes V'_F for D' according to Definition 4.*

Then there exists for every solution (q, π, s) to MINLP (3) another solution to MINLP (3) with the same or a lower objective function value such that

(i) $q_a = 0$ for all $a \in A'_{\text{out}}$,

(ii) for each component (V_C, E_C) of $(V'_{\text{in}}, A'_{\text{in}})$, there exists an ASTS-orientation $\tilde{D}_C = (V_C, \tilde{A}_C)$ with respect to $(V'_+ \cap V_C, V'_- \cap V_C, V'_T \cap V_C, V'_F \cap V_C)$ with

$$uv \in \tilde{A}_C \implies (q_{uv} \geq 0 \text{ and } q_{vu} = 0) \qquad \forall \{u, v\} \in E_C. \tag{7}$$

□

Proof In a first step we consider all cycles in our solution (q, π, s), i.e., the cycles in the digraph $D(A', q)$ induced by q. For cycles with length 2, we can construct, due to our assumption about the cost function in Section 2, a solution q' to (3) with lower or equal objective function value such that $q'_{uv} = 0$ or $q'_{vu} = 0$ for all $\{u, v\} \in E$. Regarding cycles with length greater than 2, if such a cycle is in A', it does not contain any arc from A_{gen} as A' is ASTS-representable. Due to property (2) of the feasible set F_e^{dec} and constraint (3c), the arcs on the cycle cannot contain arcs from A_{dec} either. Accordingly each arc on the cycle is in A_{mnt}. Due to A' being ASTS-representable, there is no cycle with negative total cost in $A'[E_{\text{mnt}}]$ and the lower bound for the flow on each arc in $A'[E_{\text{mnt}}]$ is 0. Therefore we can reduce the flow along each cycle such that the flow on one arc of the cycle becomes 0, without increasing the objective function value and without affecting feasibility regarding the potential bounds, since flows on arcs in A_{mnt} do not affect the potential. As a consequence, this new solution q'' to MINLP (3) is acyclic on A'.

Second, by construction of A'_{out} in Definition 2 every arc $a \in A'_{\text{out}}$ is in a block where all nodes with the possible exception of the cut-node u_c are transshipments nodes, i.e., in V'_T. Hence the total flow into or out of the block via u_c is zero, implying that any positive flow on arcs of the block is a union of cycles. By the same argument as above, we can reduce the flows on all arcs of the block to 0. Applying this to all blocks constituting A'_{out}, we obtain a solution q''' to MINLP (3) with lower or equal objective function value for which statement (i) of the theorem holds. Note that q''' is still acyclic on A'.

Third, let (V_C, A_C) be a component of $(V'_{\text{in}}, A'_{\text{in}})$ and (V_C, E_C) its underlying graph. Then (V_C, E_C) has an ASTS-orientation by construction of $(V'_{\text{in}}, A'_{\text{in}})$ due to Corollary 1 applied to $(V'_{\text{in}}, A'_{\text{in}})$. Let $\tilde{D}_{>0,C} = (V_{>0,C}, \tilde{A}_{>0,C}) := D(E_C, q''')$ be the digraph induced by q''' and $(V_{>0,C}, E_{>0,C})$ the underlying subgraph. If $V_{>0,C} = \varnothing$, any arbitrary ASTS-orientation of (V_C, E_C) trivially satisfies (7), so let us assume $V_{>0,C} \neq \varnothing$ for the remainder of this proof. Since $q'''_{uv} = 0$ or $q'''_{vu} = 0$ for all $\{u, v\} \in E$ the digraph $\tilde{D}_{>0,C}$ is in fact an orientation of $(V_{>0,C}, E_{>0,C})$. We observe that (7) holds on $\tilde{A}_{>0,C}$. Moreover, this orientation is certainly acyclic as q''' is acyclic. For all $v \in V_{>0,C} \cap V'_+$, Lemma 1 and (4) imply that v has an out-arc on $\tilde{A}_{>0,C}$. Analogously Lemma 1 and (4) imply that all $v \in V_{>0,C} \cap V'_-$ have in-arcs on $\tilde{A}_{>0,C}$. Finally, it follows from Lemma 1 and (4) that all $v \in V_{>0,C} \cap V'_T$ have both out- and in-arcs on $\tilde{A}_{>0,C}$. Hence $\tilde{A}_{>0,C}$ is an ASTS-orientation of $(V_{>0,C}, E_{>0,C})$ with respect to

$$(V_{>0,C} \cap V'_+, V_{>0,C} \cap V'_-, V_{>0,C} \cap V'_T, V_{>0,C} \cap V'_F).$$

Fourth, as we know that (V_C, E_C) has an ASTS-orientation by construction of $(V'_{\text{in}}, A'_{\text{in}})$ and we have just constructed an ASTS-orientation on the subgraph $(V_{>0,C}, E_{>0,C})$ of (V_C, E_C), we can extend this orientation to an ASTS-orientation $(V_C, \tilde{A}_C)$ of (V_C, E_C) with respect to

$$(V_C \cap V'_+, V_C \cap V'_-, V_C \cap V'_T, V_C \cap V'_F)$$

due to the completion characterization of Theorem 1. Since $q_{uv} = q_{vu} = 0$ for all edges $\{u, v\} \in E_C \setminus E_{>0,C}$, i.e., the edges oriented during this extension process, (7) now holds on all of $(V_C, \tilde{A}_C)$. ■

Remark 2 The previous theorem has shown that for a suitable subgraph every solution to MINLP (3) corresponds to an ASTS-orientation of that subgraph. Clearly, the converse is not true since node, flow or potential bounds may prevent an ASTS-orientation from having a corresponding solution to MINLP (3). □

5 Decomposition results for ASTS orientations

The graphs underlying network optimization problems are often highly structured. In this section we explore various ways to exploit structure to describe the set of ASTS orientations for a large region of the network by the sets of ASTS orientations for smaller regions. This is useful algorithmically as the number of ASTS orientations and hence the computational effort to handle them grows exponentially[1] with the number of arcs involved.

[1]Observe that ASTS orientations can be used to solve the NP-hard longest path problem.

5.1 Decomposition based on pre-oriented edges

Computational results on real-world data indicate that the flow direction for a large share of the edges can be determined using standard preprocessing techniques. This suggests to consider ASTS orientations for the remaining edges only. To study this, we need some further concepts.

Definition 5 Let $G = (V, E)$ be a graph and $\overrightarrow{A} \subseteq A[E]$ be orientations for a given subset (i.e., $a_1 = uv \in \overrightarrow{A} \implies \nexists a_2 = vu \in \overrightarrow{A}$). The graph $G^0 = (V^0, E^0)$ that arises from removing the oriented edges from G, i.e., $E^0 := E \setminus E[\overrightarrow{A}]$, $V^0 := V[E^0]$, is called the *residual graph of* G *with respect to* $\overrightarrow{A}$. We denote the components of G^0 by $G_i = (V_i, E_i)$. The set of nodes $\overline{V_i}$ in V_i that are incident to arcs in $\overrightarrow{A}$, i.e., the set

$$\overline{V_i} := \{u \in V_i \mid \exists a = uv \in \overrightarrow{A} \text{ or } \exists a = vu \in \overrightarrow{A}\}, \tag{8}$$

is called the *boundary of the component* G_i. An orientation $\tilde{D}_i = (V_i, \tilde{A}_i)$ of G_i induces, by virtue of

$$\overline{A}_i := \{uv \mid u \neq v \in \overline{V_i},\ \exists u\text{-}v\text{-path on } \tilde{D}_i\}, \tag{9}$$

a digraph $\overline{D_i} = (\overline{V_i}, \overline{A}_i)$ on the boundary $\overline{V_i}$, called the *skeleton of* $\tilde{D}_i$, that indicates the mutual reachability of nodes in the boundary of G_i in the orientation $\tilde{D}_i$. □

Remark 3 Note that a skeleton may have, for some $u, v \in V$, both arcs uv and vu. Clearly, such a skeleton is not a skeleton of an ASTS-oriented $\tilde{D}_i$. □

The next concept describes ASTS-orientations on the components G_i that are compatible with the arcs in $\overrightarrow{A}$. The idea here is that the existing arcs in $\overrightarrow{A}$ contribute to satisfying the STS-condition at the nodes of G. As a consequence, when taking into account the arcs in $\overrightarrow{A}$ we need fewer restrictions on the orientations of the components G_i to produce an ASTS-orientation of G. For a transshipment node in some component G_i with an out-arc in $\overrightarrow{A}$, for example, the STS-condition is already satisfied when there is an in-arc of the transshipment node on D_i, i.e., it is sufficient to classify the transshipment node as a sink on D_i if we would like to satisfy the STS-condition at this node on D. Similarly, to provide another example, a source on G with an out-arc in $\overrightarrow{A}$ can be classified as a free node on its component G_i if we would like to ensure that the STS-condition is satisfied on D. In this way we can reduce the problem of constructing an ASTS-orientation on a graph with a set of pre-oriented edges $\overrightarrow{A}$ to the general problem of constructing an ASTS-orientation on a graph.

Definition 6 Let $G = (V, E)$ be a graph with node subsets (V_+, V_-, V_T, V_F) and let $\overrightarrow{A}$ be a given subset of edge orientation as above, $G_i = (V_i, E_i)$ the components of the residual graph of G with respect to $\overrightarrow{A}$, and $\overline{V_i}$ the boundaries of G_i. An orientation $\tilde{D}_i = (V_i, \tilde{A}_i)$ of G_i is called an *ASTS-orientation relative to* $\overrightarrow{A}$ if it is an ASTS-orientation of G_i with respect to $(V^i_+, V^i_-, V^i_T, V^i_F)$ given by

$$V_+^i := (V_+ \cap V_i) \setminus \overline{V_i} \cup \{u \in \overline{V_i} \cap (V_+ \cup V_T) \mid \exists a = vu \in \overrightarrow{A},\, \nexists a = uw \in \overrightarrow{A}\} \tag{10a}$$

$$V_-^i := (V_- \cap V_i) \setminus \overline{V_i} \cup \{u \in \overline{V_i} \cap (V_- \cup V_T) \mid \exists a = uv \in \overrightarrow{A},\, \nexists a = wu \in \overrightarrow{A}\} \tag{10b}$$

$$V_T^i := (V_T \cap V_i) \setminus \overline{V_i}, \tag{10c}$$

$$V_F^i := V_i - V_+^i - V_-^i - V_T^i. \tag{10d}$$

□

We can now give a characterization of ASTS-orientations on G that respect the given subset $\overrightarrow{A}$ of oriented edges.

Theorem 3 *Let $G = (V, E)$ be a graph with node subsets (V_+, V_-, V_T, V_F) and $\overrightarrow{A}$, G_i, $\overline{V_i}$ be as in the preceding definition. Moreover, let $\tilde{D}_i = (V_i, \tilde{A}_i)$ be orientations of the components G_i. Then the overall digraph $\tilde{D} = (V, \overrightarrow{A} \cup \bigcup \tilde{A}_i)$ is an ASTS-orientation of G with respect to (V_+, V_-, V_T, V_F) if and only if*

(i) Each $\tilde{D}_i$ is an ASTS-orientation relative to $\overrightarrow{A}$,

(ii) the graph on V that arises when the arcs in $\overrightarrow{A}$ are added to the union of the skeletons of the orientations $\tilde{D}_i$, i.e., the graph

$$\overline{D} := (\bigcup \overline{V_i} \cup V \setminus \bigcup V_i, \overrightarrow{A} \cup \bigcup \overline{A}_i), \tag{11}$$

is acyclic, and

(iii) the nodes in $V \setminus \bigcup V_i$, i.e., the nodes of G that are only incident to arcs in $\overrightarrow{A}$, satisfy the STS-condition with respect to (V_+, V_-, V_T, V_F). □

Proof With the concepts defined above, the proof is a mere technical exercise.

$\Longrightarrow$ Let the overall digraph $\tilde{D} = (V, \overrightarrow{A} \cup \bigcup \tilde{A}_i)$ be an ASTS-orientation. We will show that the three conditions of the proposition are true.

(i) Since all nodes of $\tilde{D}$ satisfy the STS-condition, so do all nodes of the orientations $\tilde{D}_i$ that are not incident to any arc in $\overrightarrow{A}$, i.e., all nodes in $V_i \setminus \overline{V_i}$. For the nodes in $\overline{V_i}$, i.e., the nodes on the boundaries of the components G_i, we will look at the four sets $V_+^i \cap \overline{V_i}, V_-^i \cap \overline{V_i}, V_T^i \cap \overline{V_i}$ and $V_F^i \cap \overline{V_i}$ separately. Nodes in $V_+^i \cap \overline{V_i}$ are, by definition, also in $V_+ \cup V_T$, i.e., they have an out-arc on $\tilde{D}$. However, they have, also by definition, no out-arc in $\overrightarrow{A}$. Hence the out-arc must be in $\tilde{A}_i$, i.e., nodes in $V_+^i \cap \overline{V_i}$ are sources on $\tilde{D}_i$. With a similar line of argument we can see that all nodes in $V_-^i \cap \overline{V_i}$ are sinks on $\tilde{D}_i$. As nodes in V_T^i are not in the boundary of G_i, the set $V_T^i \cap \overline{V_i}$ is empty, and, finally, nodes in $V_F^i \cap \overline{V_i}$ are free nodes, i.e., they satisfy the STS-condition automatically. Altogether, we have established that all nodes in $\bigcup \overline{V_i}$ satisfy the STS-condition. As $\tilde{D}$ is acyclic, the subgraphs $\tilde{D}_i$ must be acyclic, too. Hence the digraphs $\tilde{D}_i$ are ASTS-orientations with respect to $(V_+^i, V_-^i, V_T^i, V_F^i)$.

(ii) By definition of the skeletons $\overline{\tilde{D}_i}$, the nodes of $\overline{D}$ are reachable by the same nodes in $\bigcup \overline{V_i}$ as they are on $\tilde{D}$. As a consequence, the acyclicity of $\tilde{D}$ implies the acyclicity of $\overline{D}$.

(iii) As $\tilde{D}$ is an ASTS-orientation, the nodes that are only incident to arcs in $\overrightarrow{A}$ satisfy the STS-condition. Moreover, as $(V, \overrightarrow{A})$ is a subgraph of $\tilde{D}$, it must be acyclic.

$\Longleftarrow$ We assume the three conditions are true. Then $\tilde{D}$ is acyclic because the orientations $\tilde{D}_i$ and the graph $\overline{D}$ are acyclic (conditions (i) and (ii)). We can see directly from the definition of the sets V_+^i, V_-^i, V_T^i and V_F^i that the nodes in $\bigcup V_i \setminus \overline{V_i}$ satisfy the STS-condition on $\tilde{D}$ due to condition (i), while the nodes in $V \setminus \bigcup V_i$ satisfy the STS-condition on $\tilde{D}$ due to condition (iii) of the proposition. It remains to show that the nodes in $\bigcup \overline{V_i}$ satisfy the STS-condition on $\tilde{D}$. We will look at the nodes in the sets $V_+ \cap \overline{V_i}, V_- \cap \overline{V_i}, V_T \cap \overline{V_i}$ and $V_F \cap \overline{V_i}$ separately. The nodes in $V_+ \cap \overline{V_i}$ are in V_+^i if they have an in-arc in $\overrightarrow{A}$, but no out-arc in $\overrightarrow{A}$. Due to condition (i), these nodes have an out-arc on $\tilde{D}_i$, i.e., they satisfy the STS-condition on $\tilde{D}$. The nodes in $V_+ \cap \overline{V_i}$ are in V_F^i if they have an out-arc in $\overrightarrow{A}$. In this case they satisfy the STS-condition on $\tilde{D}$ automatically. For the nodes in $V_- \cap \overline{V_i}$ a similar line of arguments applies. The nodes in $V_T \cap \overline{V_i}$ are in V_+^i, V_-^i, or V_F^i depending on the arcs in $\overrightarrow{A}$ that are incident to them. If they have an out-arc in $\overrightarrow{A}$, but no in-arc in $\overrightarrow{A}$, they are in V_-^i. Then condition (i) ensures that they have an in-arc on $\tilde{D}_i$. Accordingly, they satisfy the STS-condition on $\tilde{D}$. Analogously, condition (i) ensures that also the nodes in $V_T \cap \overline{V_i}$ that are in V_+^i fulfill the STS-condition on $\tilde{D}$. The remaining nodes in $V_T \cap \overline{V_i}$ are in V_F^i because they have both an out-arc in $\overrightarrow{A}$ and an in-arc in $\overrightarrow{A}$, which is why they satisfy the STS-condition on $\tilde{D}$ ab initio. Finally, the nodes in V_F trivially satisfy the STS-condition on $\tilde{D}$. ∎

5.2 Decomposition into blocks

The components arising from a set of pre-oriented edges $\overrightarrow{A}$ according to Definition 5 may still be rather large. However, as graphs for real-world networks are rather sparse they may be decomposed further. A natural next step is to consider decompositions into blocks.

Theorem 4 *Let $G = (V, E)$ be a graph with node set V, $V_+ \subsetneq V$, $V_- \subsetneq V$ and $V_T \subsetneq V$ disjoint sets of sources, sinks and transshipment nodes, respectively, and $V_F := V - V_+ - V_- - V_T$ the set of free nodes. Furthermore, let V_c be the set of all cut-nodes of G, $\mathcal{B}$ the set of all blocks of G, and $\tilde{D}_B = (V_B, \tilde{A}_B)$ orientations of the blocks $B \in \mathcal{B}$. Then the digraph $\tilde{D} = (V, \bigcup_{B \in \mathcal{B}} \tilde{A}_B)$ is an ASTS-orientation of G with respect to (V_+, V_-, V_T, V_F) if and only if*

(i) for all $B \in \mathcal{B}$ the orientations $\tilde{D}_B$ are ASTS-orientations with respect to

$$((V_+ \cap V_B) \setminus V_c, (V_- \cap V_B) \setminus V_c, (V_T \cap V_B) \setminus V_c, (V_F \cap V_B) \cup V_c), \text{ and}$$

(ii) the cut-nodes V_c of G satisfy the STS-condition with respect to (V_+, V_-, V_T, V_F). □

Proof $\Longrightarrow$ If $\tilde{D} = (V, \bigcup_{B\in\mathcal{B}} \tilde{A}_B)$ is an ASTS-orientation with respect to (V_+, V_-, V_T, V_F), the digraphs $(V_B, \tilde{A}_B)$ are acyclic. As the nodes in V_+, V_- and V_T satisfy the STS-condition on $\tilde{D}$, the nodes in $(V_+ \cap V_B)\setminus V_c$, $(V_- \cap V_B)\setminus V_c$ and $(V_T \cap V_B)\setminus V_c$ satisfy the STS-conditions on $\tilde{D}_B$ for sources, sinks and transshipment nodes, respectively, and condition (ii) holds. The nodes in $((V_F \cap V_B) \cup V_c)$ satisfy the STS-condition for free nodes trivially.

$\Longleftarrow$ Due to condition (i), for all $B \in \mathcal{B}$ the digraphs $\tilde{D}_B$ are acyclic, hence the tree property of the block-cut tree implies that $(V, \bigcup_{B\in\mathcal{B}} \tilde{A}_B)$ is acyclic, too. As the nodes in $(V_+ \cap V_B) \setminus V_c, (V_- \cap V_B) \setminus V_c, (V_T \cap V_B) \setminus V_c$ and $(V_F \cap V_B) \setminus V_c$ satisfy the STS-conditions for sources, sinks, transshipment nodes and free nodes, respectively, for all $B \in \mathcal{B}$, so do all nodes in $V \setminus V_c$. The remaining nodes in V_c satisfy the STS-condition due to condition (ii). ■

The previous theorem provided a block decomposition for any ASTS-orientation of a given graph G. However, when solving MINLP (3) it is sufficient to consider only those ASTS-orientations that exist according to Theorem 2 and are compatible with a solution to MINLP (3) in the sense of (7). Moreover, the fact that decomposing a graph into blocks reveals the global tree structure of the graph (as captured in the block-cut tree of the graph) provides additional information about the flows through cut-nodes. Via a suitable definition of sources, sinks, transshipment nodes and free nodes relative to a block, this information further reduces the number of ASTS-orientations we need to consider for finding a solution to (3).

More precisely speaking, the flow through a cut-node into a block B is entirely determined by the nodes in the blocks of the branch of the block-cut tree that is connected with the node after removing B from the block-cut tree. The following definition provides the necessary concepts.

Definition 7 In the setting of Theorem 2 let $C := (V_C, E_C)$ be a component of $(V'_{\text{in}}, A'_{\text{in}})$, V_c the set of all cut-nodes of C, $\mathcal{B}$ the set of all blocks of C, T the block-cut tree of C, and for all blocks $B = (V_B, E_B) \in \mathcal{B}$ let $T_B := T - \{B\}$ be the forest we obtain by removing from T the node that respresents the block B. For a cut-node v of a block B, i.e. $v \in V_c \cap V_B$, we define $G_{B,v}$ to be the subgraph of C that is the union of all blocks in the component of T_B that contains v. We call the node set $V(G_{B,v})$ the *flow determining nodes of v relative to B* and

$$V_{+,B} := ((V'_+ \cap V_B) \setminus V_c) + \{v \in V_c \cap V_B \mid \sum_{u\in V(G_{B,v})} \underline{q}_u^{A'} > 0\}, \tag{12a}$$

$$V_{-,B} := ((V'_- \cap V_B) \setminus V_c) + \{v \in V_c \cap V_B \mid \sum_{u\in V(G_{B,v})} \overline{q}_u^{A'} < 0\}, \tag{12b}$$

$$V_{T,B} := ((V'_T \cap V_B) \setminus V_c) + \{v \in V_c \cap V_B \mid \sum_{u\in V(G_{B,v})} \underline{q}_u^{A'} = \sum_{u\in V(G_{B,v})} \overline{q}_u^{A'} = 0\}, \tag{12c}$$

$$V_{F,B} := V_B - V_{+,B} - V_{-,B} - V_{T,B}. \tag{12d}$$

the *sources, sinks, transshipment nodes and free nodes relative to B.* □

The following theorem provides the block decomposition result when we take into account flows compatible with MINLP (3). Moreover, it allows us to further reduce the number of ASTS-orientations considered by indentifying blocks that necessarily have flow zero.

Theorem 5 *In the setting of Theorem 2 let* $B = (V_B, E_B) \in \mathcal{B}$ *be a block of a component* $C = (V_C, E_C)$ *of* $(V'_{\text{in}}, A'_{\text{in}})$*. Consider an ASTS-orientation* $\tilde{D} = (V_C, \tilde{A}_C)$ *of* C *corresponding to the flow part* q *of a solution to MINLP* (3) *satisfying* (7) *as constructed in the proof of Theorem 2.*

(i) If B *does not have at least one in-node and a distinct out-node, i.e.,*

$$|V_{+,B} \cup V_{F,B}| = 0 \quad \text{or} \quad |V_{-,B} \cup V_{F,B}| = 0 \quad \text{or} \quad |V_{+,B} \cup V_{-,B} \cup V_{F,B}| \leq 1 \tag{13}$$

then $q_a = 0$ *for all arcs* $a \in A[E_B]$.

(ii) For all $B \in \mathcal{B}$ *that do not satisfy* (13) *there exist ASTS-orientations* $\tilde{D}_B = (V_B, \tilde{A}_B)$ *with respect to* $(V_{+,B}, V_{-,B}, V_{T,B}, V_{F,B})$ *such that* $\bigcup_{B \in \mathcal{B}} \tilde{A}_B$ *is equal to* $\tilde{A}_C$ *except for arcs with flow zero, i.e.*

$$\{a \in \bigcup_{B \in \mathcal{B}} \tilde{A}_B \mid q_a \neq 0\} = \{a \in \tilde{A}_C \mid q_a \neq 0\}. \tag{14}$$

□

Proof Let $B = (V_B, E_B)$ be a block of (V_C, E_C). We start by showing that in analogy to Lemma 1, the sets $V_{+,B}$, $V_{-,B}$, and $V_{T,B}$ are the sources, sinks, and transshipment nodes of the block B for every flow solution. Formally, for a node $u \in V_{T,B}$ we show that for every flow q that is feasible for MINLP (3), there is an in-arc $vu \in A_B$ with $q_{vu} > 0$ iff there is an out-arc $uw \in A_B$ with $q_{uw} > 0$. If the node u is not a cut-node, this follows immediately from Definition 7 and Lemma 1. In case u *is* a cut-node, the flow into B via u is given by the sum of the inflows provided by the nodes $V(G_{u,B})$. According to (12c), this sum is 0. Hence flow through u happens only via arcs in B. By flow conservation there is an in-arc carrying flow iff and only if there is a corresponding out-arc. The same reasoning can be used to show analogous statements for $V_{+,B}$ and $V_{-,B}$. We have thus shown that the STS-condition is satisfied for the underlying graph of the flow-induced digraph $D(E_B, q)$.

To establish (i) consider a block B satisfying (13). If $|V_{+,B} \cup V_{-,B} \cup V_{F,B}| = 0$, positive flow in B can happen only in cycles. Due to the construction in the proof of Theorem 2, the flows have been reduced to 0 in this case without violating feasibility or increasing the cost. If $|V_{+,B} \cup V_{-,B} \cup V_{F,B}| \geq 1$, but either $|V_{+,B} \cup V_{F,B}| = 0$ or $|V_{-,B} \cup V_{F,B}| = 0$ there is either no source or no sink for the flow, which is not possible due to flow conservation. Therefore this case is excluded.

To show (ii), let B be a block of (V_C, A_C) that does not satisfy (13). We already know that the STS-condition holds for the underlying graph $(V_{>0,B}, E_{>0,B})$ of the flow-induced digraph $D(E_B, q)$. If $V_{>0,B} = \emptyset$, any ASTS orientation on B with respect to $(V_{+,B}, V_{-,B}, V_{T,B}, V_{F,B})$ trivially satisfies (14). Such an orientation must exist due to Corollary 1 since B does not satisfy (13). In case $V_{>0,B} \neq \emptyset$ the flow-induced digraph $D(E_B, q)$ is acyclic as it is a subgraph of $\tilde{D}$ and hence it is in fact an ASTS-orientation with respect to $(V_{+,B}, V_{-,B}, V_{T,B}, V_{F,B})$. We can extend the ASTS orientation of $(V_{>0,B}, E_{>0,B})$ to an orientation $\tilde{D}_B$ of B due to the completion characterization of Theorem 1. As all orientations $\tilde{D}_B$ constructed in this fashion certainly satisfy (14), we have proved (ii).

Remark 4 1. The bounds for the node demands calculated in the previous Theorem 5 (and the classification of nodes into sources, sinks, transshipment nodes and free nodes building of these bounds) can be tightened significantly boundary combining the information about node demand bounds resulting from considering the block-cut trees of the components of the subgraph (V'_{in}, A'_{in}) (as in the previous theorem) with node demand bounds that can be obtained from considering the block-cut tree of the subgraph (V_{in}, A_{in}).

2. Algorithmically, it is faster to calculate $V_{+,B}$, $V_{-,B}$ and $V_{T,B}$ recursively starting from the leaves of the block-cut tree than to sum up the demand bounds of all nodes of the blocks of the subgraphs $G_{B,v}$ (as in Theorem Theorem 5). □

6 Exploiting ASTS orientations algorithmically

Theorem 2 implies that given a suitable ASTS-representable region A', the MINLP (3) complemented by the constraints

$$q_a = 0 \quad \text{for all } a \in A'_{\text{out}}, \tag{15a}$$
$$\{D(A', q) \text{ is an ASTS-orientation w.r.t. } (V_+, V_-, V_T, V_F)\} \tag{15b}$$

is equivalent to (3) in the sense that one of the two problems is feasible iff the other one is feasible and both problems have the same objective value. To formulate (15b) as a MILP constraint, we first introduce binary variables x_e for the flow direction of each edge $e = \{u, v\} \in E'_{\text{in}} := E[A'_{\text{in}}]$, where $x_e = 1$ means flow along uv and $x_e = 0$ indicates flow along vu. The coupling between these binary variables and the corresponding flow variables can be achieved via the big-M constraints

$$q_{uv} \leq x_e \overline{q}_{uv} \qquad \text{and} \qquad q_{vu} \leq (1 - x_e)\overline{q}_{vu}. \tag{16}$$

To model the requirements for an ASTS orientation, we employ a Dantzig-Wolfe approach: We (conceptually) enumerate all ASTS orientations and require that one of those is chosen. Of course, this approach will not work for large networks as we expect the number of ASTS orientations to grow exponentially in the number of arcs. For this reason, we consider a family $\mathcal{A} = \{A'_1, \dots, A'_k\}$ of ASTS-representable regions for which all ASTS orientations can be enumerated with reasonable effort. Observe that in general this does not guarantee that an ASTS orientation for the maximal ASTS-representable arc subset $\hat{A}$ is chosen. Let $\mathcal{O}_i$, $1 \leq i \leq k$, be the set of ASTS orientations for the edge set $E'_{i,\text{in}}$ corresponding to region A'_i. For each $\tilde{D}_j = (V_j, \tilde{A}_j) \in \mathcal{O}_i$ we introduce a binary variable $y^C_{\tilde{D}}$, where $\tilde{D}$ is selected iff $y^C_{\tilde{D}} = 1$. This requirement is formulated as

$$\sum_{\tilde{D}_j \in \mathcal{O}_i \colon uv \in \tilde{A}_j} y^C_{\tilde{D}} = x_e \quad \text{for all } e = \{u, v\} \in E'_{i,\text{in}}. \tag{17}$$

Adding constraints (15a), (16), and (17) for each region A'_i can be used to strengthen the MINLP model (3) or any relaxation of it. In particular, it can be used to strengthen the classical network flow subproblem of (3), which in turn can be used in an optimality-based bound tightening procedure (see Section 7.3). To this end, the resulting MILP model is minimized and maximized for each arc flow variable in turn to obtain stronger flow bounds.

The framework presented so far offers many algorithmic opportunities. We propose the following approach aiming to

- determine as many zero flow arcs as possible,
- to forbid as many non-ASTS-orientations as possible, and
- to obtain small arc subsets for which ASTS-orientations are enumerated.

Note that the last two goals are obviously in conflict with each other.

The approach consists of the following steps:

1. Apply Theorem 2 to the maximal ASTS-representable arc subset $\hat{A}$ to determine the set $\hat{A}_{\text{out}}$ of arcs with zero flow and the inner arcs $\hat{A}_{\text{in}}$.
2. Apply Theorem 5 to $\hat{A}_{\text{in}}$ to obtain an additional set $\hat{A}_{\text{in},0}$ of zero flow arcs according to (i) and a set of blocks $\hat{\mathcal{B}}$ according to (ii).
3. Within each block $B \in \hat{\mathcal{B}}$, consider the components G_i of the residual graph of B with respect to the pre-oriented arcs in B according to Theorem 3.
4. For each component G_i of each block $B \in \hat{\mathcal{B}}$, apply Theorem 2 and Theorem 5 to obtain further sets of zero flow arcs and a set of blocks $\mathcal{B}_i$.
5. Enumerate the ASTS-orientations of each block in
$$\bigcup_{G_i \text{ is component of } B \in \hat{\mathcal{B}}} \mathcal{B}_i$$
separately to set up the configuration model (17).

Observe that using the ASTS-orientations for the blocks obtained in step 2 in the configuration model (17) is exact, i.e., forbids all non-ASTS orientations. We obtain an relaxation of the restriction to ASTS-orientations of $\hat{A}$ as the constraints described so far only enforce conditions (i) and (iii) of Theorem 3, but not (ii), the requirement that the resulting orientation is acyclic. Of course, this could be enforced too, e.g., by explicitly forbidding combinations of orientations leading to a cycle. However, we do not consider this in this paper.

For our computational proof of concept in Section 7 we will employ a somewhat lighter approach in the following, which is based on the ideas of Corollary 1, Theorem 3 and Theorem 4.

1. Apply Theorem 2 to the maximal ASTS-representable arc subset $\hat{A}$ to determine the set $\hat{A}_{\text{out}}$ of arcs with zero flow and the inner arcs $\hat{A}_{\text{in}}$.
2. Consider the components of the subgraph induced by $\hat{A}_{\text{in}}$ consisting of those edges for which the flow bounds do not yet determine the flow direction.

Consider the edge set of each of these components as a region and determine the local inner and outer edge sets A'_{in} and A'_{out} according to Theorem 2. Decompose the edge sets A'_{in} further into blocks and enumerate the ASTS-orientations of each of those blocks separately to set up the configuration model (17).

As above, this scheme does not exclude cycles through multiple regions (cf. (ii) of Theorem 3). However, there is already a significant computational impact as we will see in the next section.

7 Computational results

In order to investigate the computational potential of these ideas we consider gas network optimization instances used in the literature [KHPS15, SAB$^+$17]. We study the improvement in the bound tightening over existing problem-specific bound tightening techniques as well as the impact of strengthening optimization models by information due to ASTS orientations on the running time to solve the models.

7.1 Test instances

In order to benefit from analyzing possible ASTS orientations, the network has to feature several cycles. We therefore use the largest networks of the public GasLib [SAB$^+$17], GasLib-582 (version 2) and GasLib-4197, containing many cycles. These are complemented by non-public data for a real-world network HN-AB that has also been studied in [PFG$^+$15, KHPS15]. Although the networks GasLib-582 and HN-AB are of comparable size it has been observed [PFG$^+$15] that instances for HN-AB are harder to solve.

MINLP models for networks of this size (see Table 1) cannot be solved in reasonable time by current solvers [PFG$^+$15]. We therefore consider the MILP relaxation [GMMS15, GMMS12] of a certain gas network MINLP that has been successfully used to solve the GasLib-582 and HN-AB instances in [KHPS15].

In order to apply the theory we developed, we treated the network elements pipes and control valves as potential-decreasing edges, the network elements valves, resistors, and shortpipes as potential-maintaining edges, and compressor elements as generic edges. To match the MINLP model to these choices, we modified each network in the following way. First, the altitude of each network node is set to 0, hence all pipes are horizontal. Thus the potential drop function for a pipe is $\phi_a(q_a) = c\phi_a|\phi_a|$ for some constant c, fulfilling the requirement (2). Moreover, we replaced resistors by open valves. With this modified data, the MILP from [GMMS15] is a relaxation of a MINLP of the type (3).

For each instance (consisting essentially of a gas demand vector) we use the lamatto framework [Lam14] that has also been used in [GMMS12, GMMS15, GMSS15, GMSS18] to generate the MILP relaxation. It is crucial to note that this generation process includes a state-of-the-art bound tightening procedure summarized in [SKMP15]. Among other bound tightening techniques, this includes optimality-based bound tightening for the flows, i.e., minimizing and maximizing the flow over each arc subject to flow conservation constraints and flow bounds (either from the original input or derived via other bound tightening steps). This bound tightening is able to fix the flow direction on a large

instance set		# arcs		# instances
	decreasing	maintaining	increasing	
GasLib-582	301	303	5	3545 (4227)
HN-AB	524	158	7	42 (43)
GasLib-4197	3657	797	12	2014 (2859)

Table 1: Statistics for the considered instance sets. The column "# instances" gives the number of "not infeasible" instances as described above and, in parantheses, the number of all instances.

share of the network arcs, but there also remain large parts of the network where the flow direction cannot be fixed. As an objective, we chose to minimize the total amount of compressed gas, i.e., the sum of the flow variables through active compressor stations. The root node of each MILP model was solved with Gurobi 8.1 [GO19]; in the following, we consider only those instances which are not infeasible after the root node.

7.2 Implementation and computational setup

The algorithms have been coded prototypically in Python. For the enumeration of ASTS orientations, a simple backtracking search is used. The enumeration is stopped as soon as at least 2000 ASTS orientations have been generated. For components with so many orientations, no orientations are considered in the following to avoid an unreasonable blowup of the MILP model used for OBBT as well the original MILP extended by the configuration model based on the enumerated orientations.

To measure the tightness of flow bounds for a network arc, we define the *flow range* to be the difference between the upper and the lower flow bound of that arc. In order to have an instance-independent tightness measure, the *relative flow range* is the flow range divided by *twice* the total inflow, i.e., the sum of the flows entering the network. Hence the relative flow range is in the interval $[0, 1]$. We use this to limit computation time for OBBT: the flow bounds for a variable are tightened only if the relative flow range is at least 2.5%. The MILP models during OBBT are solved using SCIP 6.0 [GBE+18].

All computations were performed on machines with Intel Xeon E5-2670 CPUs with 2.5 GHz and 64 GB of RAM. The runtimes reported are for single threaded computations that use the machine exclusively.

7.3 Strengthening flow bounds

We start our evaluation by investigating the improvement of the bounds for the flow variables when performing OBBT using the MILP model consisting of the classical network flow constraints (3b), (3h), and (3i) complemented by the configuration models (17) for choosing an ASTS orientation in each selected region. As a benchmark, we compare against the bounds obtained by the lamatto bound tightening algorithm (described in [SKMP15]), in the following refered to as "lamatto BT". This algorithms performs, among others, classical bound tightening for the constraints of the network elements, as well as OBBT using

flow range	lamatto BT	OBBT with orientations	rel. improvement [%]
== 0.0	2739	2779	1.5
≥ 1000.0	334	223	33.2
≥ 2000.0	268	207	22.8
≥ 3000.0	190	140	26.3
≥ 4000.0	134	6	95.5
≥ 5000.0	130	0	100.0
≥ 6000.0	56	0	100.0

Table 2: Comparison of flow range distribution for GasLib-4197 instance `nomination_mild_1280.lp`. The first row gives the number of arcs for which the flow value has been fixed.

the classical network flow problem. Our bound tightening procedure, "OBBT with orientations", performs OBBT using the MILP model explained above for each arc (with sufficiently high relative flow range, see above) starting from the flow bounds obtained by "lamatto BT".

To measure the strength of the flow bounds, we consider the distribution of the flow range values for each instance. We compare the flow range distribution of "lamatto BT" vs. "OBBT with orientations" by comparing their tails, i.e., for how many arcs the flow range exceeds certain thresholds. An example for the GasLib-4197 instance `nomination_mild_1280.lp` is shown in Table 2. For instance, using "lamatto BT" there remain 56 arcs with a flow range of at least 6000 units, whereas "OBBT with orientations" reduces the flow range of all arcs below 5000 units.

We extend this kind of analysis to all instances of an instance set by considering for each instance the relative flow range as defined above. This allows us to use fixed thresholds for all instances; these thresholds are relative to the instance-specific total inflow. The results are shown in Table 3. It is evident that "OBBT with orientations" is very effective in improving the bounds for arcs with a large initial flow range. Moreover, for 1% to 2% additional arcs the flow can actually be fixed. Note that the bounds could be tightened further by iterating the lamatto bound tightening and our OBBT using ASTS orientations until no further improvement is possible.

7.4 Improved models for gas network operation

To investigate the computational impact of exploiting information derived from ASTS orientations, we consider the following three MILP variants:

"original" MILP model as generated using the lamatto framework

"original+bounds" extends the "original" model by bounds obtained via OBBT using feasible ASTS orientations

"original+bounds+orientations" extends "original+bounds" further by the configuration model (17) for these orientations

	min	0.25 quantile	median	0.75 quantile	max
== 0.0	0.0	0.0	2.2	2.2	3.3
$\geq$ 0.1	10.9	16.4	16.4	16.4	19.3
$\geq$ 0.2	10.9	11.5	11.5	16.4	19.3
$\geq$ 0.3	10.9	11.5	11.5	11.5	32.5
$\geq$ 0.4	10.9	11.5	11.5	11.5	32.5
$\geq$ 0.5	19.0	29.4	29.4	31.7	69.1
$\geq$ 0.6	35.3	47.6	47.6	47.6	58.3
$\geq$ 0.7	35.3	47.6	52.4	52.4	52.4
$\geq$ 0.8	35.3	41.2	41.2	52.4	68.8
$\geq$ 0.9	35.3	41.2	45.2	58.3	75.9

(a) GasLib-582

	min	0.25 quantile	median	0.75 quantile	max
== 0.0	0.0	0.0	0.0	0.9	1.1
$\geq$ 0.1	9.8	11.8	14.6	14.6	15.0
$\geq$ 0.2	6.7	11.8	11.8	11.8	12.1
$\geq$ 0.3	6.7	11.8	11.8	11.8	15.4
$\geq$ 0.4	7.3	11.5	11.8	11.8	13.0
$\geq$ 0.5	4.1	10.3	14.5	19.1	30.2
$\geq$ 0.6	14.7	30.2	36.2	38.2	54.8
$\geq$ 0.7	16.7	30.2	36.2	38.2	56.5
$\geq$ 0.8	20.6	30.2	32.5	40.6	61.9
$\geq$ 0.9	20.6	21.2	29.2	41.9	65.0

(b) HN-AB

	min	0.25 quantile	median	0.75 quantile	max
== 0.0	1.1	1.3	1.5	1.6	2.6
$\geq$ 0.1	32.8	35.6	36.8	37.4	39.0
$\geq$ 0.2	24.4	28.5	31.8	35.1	39.1
$\geq$ 0.3	20.4	23.9	26.6	30.1	34.5
$\geq$ 0.4	20.7	23.1	24.3	27.5	35.0
$\geq$ 0.5	25.3	27.0	29.0	30.1	37.9
$\geq$ 0.6	60.4	63.6	65.0	95.5	96.4
$\geq$ 0.7	62.7	66.4	67.4	95.5	100.0
$\geq$ 0.8	55.2	66.4	66.9	100.0	100.0
$\geq$ 0.9	52.2	66.4	95.9	100.0	100.0

(c) GasLib-4197

Table 3: Comparison of the improvement of the relative flow range distribution over all considered instances of each instance set. The tables show the distribution of the reduction (in percentage of arcs) of the number of arcs whose relative flow range exceeds the given threshold. For instance, in the instance set GasLib-4197 the number of arcs with a relative flow range of at least 0.9 is reduced by at least 52.2%, 95.9% in the median, and at most 100%.

We use Gurobi 8.1 to solve these MILPs for each instance with a runtime limit of 3600 seconds. As the instances are numerically challenging, we used the Gurobi parameter `NumericFocus=3`. Each of the three MILP model variants is solved 5 times using distinct seeds; "runtime" in the subsequent evaluation refers to the average of the runtime of these 5 runs (using the time limit in case the instance has not been solved). Apart from GasLib-4197, all instances have been solved to optimality. Due to limited computational resources, we preliminarily selected a subset of 104 instances from GasLib-4197.

The results are summarized in the performance profiles [DM02] in Fig. 1. The performance profiles are restricted to "non-trivial" instances, i.e., those for which at least one of the models has a runtime of at least 10 seconds.

Only roughly one quarter of the GasLib-582 instances is "non-trivial" in this sense, showing that this instance set is rather easy to solve. The difference between the "original" and "original+bounds" models are negligible, with the performance of "original+bounds+orientations" being clearly inferior. This is expected, as the additional model complexity due to the configurations models for the orientations is overkill for the already easy instances.

For the HN-AB instance set, which is known to be harder than GasLib-582, 29 out of the 42 instances are "non-trivial". Initially, "original" and "original+bounds" perform rather similar and close to the virtual best solver, with "original+bounds+orientations" being clearly outperformed. However, on harder instances both "original+bounds" and "original+bounds+orientations" outperform "original" with "original+bounds+orientations" being initially inferior, but becoming eventually superior.

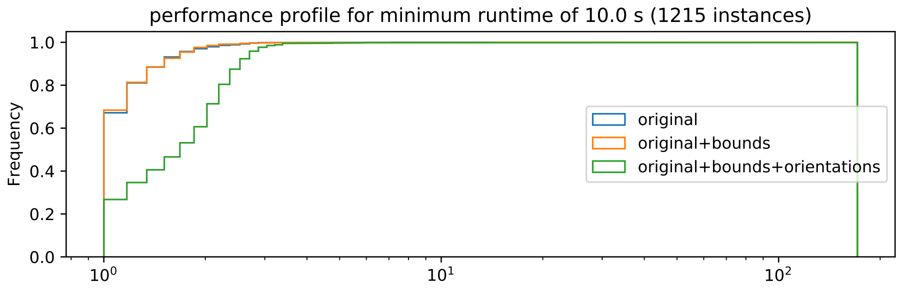
All the studied GasLib-4197 are "non-trivial" and the effect observed already for HN-AB is even more pronounced: "original+bounds" consistently outperforms "original", and is itself being outperformed by "original+bounds+orientations" for harder instances.

To summarize, our results clearly show that the harder the instances, the more beneficial it is to use the proposed extended models. We should mention, however, that with the current implementation the runtime gains are exceeded by the computation times for analysing ASTS orientations and, most importantly, for performing OBBT using the orientations.

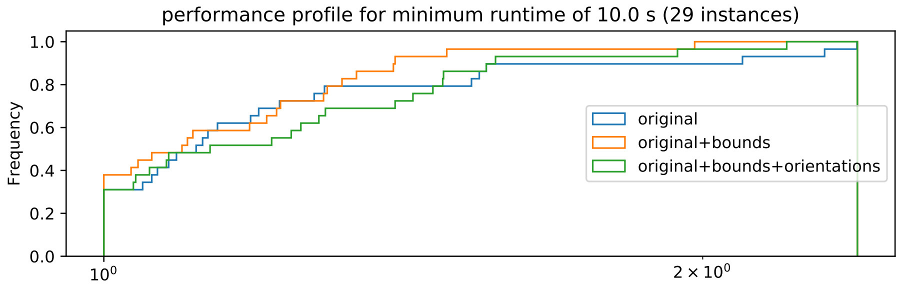
Finally, we observe that the runtime benefit reported in [HP20] for GasLib-582 is more significant than the one we found. We conjecture that this is due to the fact that we are currently using a rather coarse MILP approximation of the underlying MINLP, whereas [HP20] considers solving the MINLP directly. In other words, our benchmark problem is rather easy, leaving not as much room for improvement. This suspicion is supported by comparing the average run times for the reference models, which are 7.8 seconds for our MILP model and 1660 seconds for the MINLP model from [HP20]. We are currently working on refined computational results to investigate this issue.

8 Conclusions and further work

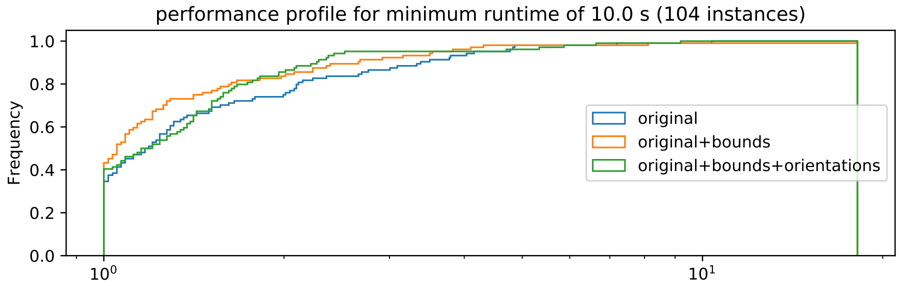
We proposed ASTS orientations as a combinatorial relaxation capturing essential properties of feasible solutions of potential-driven flow problems. Incorporating the restriction to ASTS orientations into established relaxations leads to tighter relaxations. Our computational results for large-scale gas networks



(a) GasLib-582



(b) HN-AB



(c) GasLib-4197

Figure 1: Performance profiles for the average running time of solving each MILP instance 5 times using Gurobi.

indicate that there is a significant computational advantage if the considered instances are sufficiently hard. It is interesting to note that these substantial improvements can already be gained by considering small subgraphs only.

The theoretical framework for ASTS orientations presented in this paper offers further algorithmic opportunities that have not been investigated yet. For instance, we did not yet exclude cycles encompassing several undirected components as suggested by Theorem 3. Algorithmically, this can be done by forbidding or separating such cycles via dicycle inequalities, similar to the approach in [HP20]. Another line of research is to develop efficient variants of OBBT based on ASTS orientations.

Acknowledgements This research was carried out in the framework of MATHEON supported by the Einstein Foundation Berlin. Moreover, the authors thank the BMBF Research Campus Modal (fund number 05M14ZAM) for additional support.

References

[BGS09] J. Burgschweiger, B. Gnädig, and M. C. Steinbach. Optimization models for a operative planning in drinking water networks. *Optim. Eng.*, 10:343–373, 2009.

[BH18] Kai Helge Becker and Benjamin Hiller. ASTS orientations on undirected graphs. ZIB Report 18-31, Zuse Institute Berlin, 2018. `http://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/6963`.

[BH19] Kai Helge Becker and Benjamin Hiller. Improved optimization models for potential-driven network flow problems via ASTS orientations. ZIB Report 19-58, Zuse Institute Berlin, 2019. `http://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/7534`.

[CCH+78] M. Collins, L. Cooper, R. Helgason, J. Kennington, and L. LeBlanc. Solving the pipe network analysis problem using optimization techniques. *Management Sci.*, 24(7):747–760, 1978.

[dFOdMR95] Hubert de Fraysseix, Patrice Ossona de Mendez, and Pierre Rosenstiehl. Bipolar orientations revisited. *Discrete Applied Mathematics*, 56(2-3):157–179, 1995.

[DLWB15] Claudia D'Ambrosio, Andrea Lodi, Sven Wiese, and C. Bragalli. Mathematical programming techniques in water network optimization. *European Journal of Operational Research*, 243(3):774–788, 2015.

[DM02] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Math. Programming*, 91(2):201–213, 2002.

[GBE^{+}18] Ambros Gleixner, Michael Bastubbe, Leon Eifler, Tristan Gally, Gerald Gamrath, Robert Lion Gottwald, Gregor Hendel, Christopher Hojny, Thorsten Koch, Marco E. Lübbecke, Stephen J. Maher, Matthias Miltenberger, Benjamin Müller, Marc E. Pfetsch, Christian Puchert, Daniel Rehfeldt, Franziska Schlösser, Christoph Schubert, Felipe Serrano, Yuji Shinano, Jan Merlin Viernickel, Matthias Walter, Fabian Wegscheider, Jonas T. Witt, and Jakob Witzig. The SCIP Optimization Suite 6.0. ZIB-Report 18-26, Zuse Institute Berlin, July 2018.

[GHHS16] Claudia Gotzes, Holger Heitsch, René Henrion, and Rüdiger Schultz. On the quantification of nomination feasibility in stationary gas networks with random load. *Mathematical Methods of Operations Research*, 84(2):427–457, 2016.

[GHHV12] A. M. Gleixner, H. Held, W. Huang, and S. Vigerske. Towards globally optimal operation of water supply networks. *Num. Algebra, Control and Optimization*, 2:695–711, 2012.

[GJR85] M. Grötschel, M. Jünger, and G. Reinelt. On the acyclic subgraph polytope. *Mathematical Programming*, 33(1), 1985.

[GMMS12] B. Geißler, A. Martin, A. Morsi, and L. Schewe. Using piecewise linear functions for solving MINLPs. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 287–314. Springer New York, 2012.

[GMMS15] Björn Geißler, Alexander Martin, Antonio Morsi, and Lars Schewe. The MILP-relaxation approach. In Koch et al. [KHPS15].

[GMSS15] Björn Geißler, Antonio Morsi, Lars Schewe, and Martin Schmidt. Solving power-constrained gas transportation problems using an MIP-based alternating direction method. *Computers & Chemical Engineering*, 82:303–317, 2015.

[GMSS18] Björn Geißler, Antonio Morsi, Lars Schewe, and Martin Schmidt. Solving highly detailed gas transport MINLPs: Block separability and penalty alternating direction methods. *INFORMS Journal on Computing*, 30(2):309–323, 2018.

[GNS17] Claudia Gotzes, Sabrina Nitsche, and Rüdiger Schultz. Probability of feasible loads in passive gas networks with up to three cycles. Preprint, SFB TRR 154, March 2017. Submitted.

[GO19] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2019.

[GPS^{+}19] Martin Groß, Marc E. Pfetsch, Lars Schewe, Martin Schmidt, and Martin Skutella. Algorithmic results for potential-based flows: Easy and hard cases. *Networks*, 73(3), 2019.

[HB18] Benjamin Hiller and Kai Helge Becker. Improving relaxations for potential-driven network flow problems via acyclic flow orientations. ZIB Report 18-30, Zuse Institute Berlin, 2018. http://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/6962.

[HF13] Jesco Humpola and Armin Fügenschuh. A new class of valid inequalities for nonlinear network design problems. ZIB-Report 13–06, Zuse Institute Berlin, Takustr.7, 14195 Berlin, Germany, 2013.

[HHH+15] Benjamin Hiller, Christine Hayn, Holger Heitsch, René Henrion, Hernan Leövey, Andris Möller, and Werner Römisch. Methods for verifying booked capacities. In Koch et al. [KHPS15].

[HJ84] Chris T. Hendrickson and Bruce N. Janson. A common network flow formulation for several civil engineering problems. *Civil Engineering Systems*, 1(4):195–203, 1984.

[HP20] Oliver Habeck and Marc E. Pfetsch. Combinatorial acyclicity models for potential-based flows. Technical report, TU Darmstadt, 2020. http://www.optimization-online.org/DB_HTML/2020/04/7761.html.

[HPU19] Oliver Habeck, Marc E. Pfetsch, and Stefan Ulbrich. Global optimization of mixed-integer ODE constrained network problems using the example of stationary gas transport. *SIAM J. Comput.*, 29(4):2949–2985, 2019.

[KHPS15] Thorsten Koch, Benjamin Hiller, Marc Pfetsch, and Lars Schewe, editors. *Evaluating Gas Network Capacities*. MOS-SIAM Series on Optimization. SIAM, 2015.

[Lam14] LaMaTTO++: A framework for modeling and solving mixed-integer nonlinear programming problems on networks., 2014. https://en.www.math.fau.de/edom/projects-edom/mixed-integer-programming/lamatto/.

[Mau77] J. J. Maugis. Etude de réseaux de transport et de distribution de fluide. *RAIRO Rech. Opér.*, 11(2):243–248, 1977.

[PFG+15] Marc E. Pfetsch, Armin Fügenschuh, Björn Geißler, Nina Geißler, Ralf Gollmer, Benjamin Hiller, Jesco Humpola, Thorsten Koch, Thomas Lehmann, Alexander Martin, Antonio Morsi, Jessica Rövekamp, Lars Schewe, Martin Schmidt, Rüdiger Schultz, Robert Schwarz, Jonas Schweiger, Claudia Stangl, Marc C. Steinbach, Stefan Vigerske, and Bernhard M. Willert. Validation of nominations in gas network optimization: models, methods, and solutions. *Optimization Methods and Software*, 30(1):15–53, 2015.

[Rag13] A. U. Raghunathan. Global optimization of nonlinear network design. *SIAM Journal on Optimization*, 23(1):268–295, 2013.

[SAB+17] Martin Schmidt, Denis Aßmann, Robert Burlacu, Jesco Humpola, Imke Joormann, Nikolaos Kanelakis, Thorsten Koch, Djamal Oucherif, Marc E. Pfetsch, Lars Schewe, Robert Schwarz, and Mathias Sirvent. GasLib – a library of gas network instances. *Data*, 2(40), 2017.

[SKMP15] Lars Schewe, Thorsten Koch, Alexander Martin, and Marc E. Pfetsch. Mathematical optimization for evaluating gas network capacities. In Koch et al. [KHPS15].

[VMC15] M. Vuffray, S. Misra, and M. Chertkov. Monotonicity of dissipative flow networks renders robust maximum profit problem tractable: General analysis and application to natural gas flows. In *IEEE Conference on Decision and Control (CDC)*, 2015.