Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

ARMIN FÜGENSCHUH, JESCO HUMPOLA

A Unified View on Relaxations for a Nonlinear Network Flow Problem

ZIB-Report 13-31 (July 2013)

Herausgegeben vom
Konrad-Zuse-Zentrum für Informationstechnik Berlin
Takustraße 7
D-14195 Berlin-Dahlem

Telefon: 030-84185-0
Telefax: 030-84185-125

e-mail: bibliothek@zib.de
URL: http://www.zib.de

ZIB-Report (Print) ISSN 1438-0064
ZIB-Report (Internet) ISSN 2192-7782

A Unified View on Relaxations for a Nonlinear Network Flow Problem

Armin Fügenschuh[1] and Jesco Humpola[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] Zuse Institute Berlin,
Takustr. 7, 14195 Berlin, Germany
humpola@zib.de
http://www.zib.de

Abstract. We consider a nonlinear nonconvex network flow problem that arises, for example, in natural gas or water transmission networks. Given is such network with active and passive components, that is, valves, compressors, pressure regulators (active) and pipelines (passive), and a desired amount of flow at certain specified entry and exit nodes of the network. Besides flow conservation constraints in the nodes the flow must fulfill nonlinear nonconvex pressure loss constraints on the arcs subject to potential values (i.e., pressure levels) in both end nodes of each arc. The problem is how to numerically compute this flow and pressures. We review an existing approach of Maugis (1977) and extend it to the case of networks with active elements (for example, compressors). We further examine different ways of relaxations for the nonlinear network flow model. We compare different approaches based on nonlinear optimization numerically on a set of test instances.

Keywords: Nonlinear Network Flow; Mixed-Integer Nonlinear Programming; Relaxations.

1 Introduction

We consider a nonlinear nonconvex network flow problem that arises, for example, in natural gas or water transmission networks; for a survey of other application areas we refer to Dembo et al. [13]. Given is such network with active and passive components, that is, valves, compressors, pressure regulators (active) and pipelines (passive), and a desired amount of flow at certain specified entry and exit nodes of the network. Besides flow conservation constraints in the nodes the flow must fulfill nonlinear nonconvex pressure loss constraints on the arcs subject to potential values (i.e., pressure levels) in both end nodes of each arc. The problem is how to numerically compute this flow and pressures. One possible way that leads to a global optimal solution (or a proof of infeasibility) is to use a mixed-integer nonlinear formulation, and linear programming based branched-and-bound techniques, where branching occurs on integer decisions as well as on nonlinear structures (spatial branching). The main obstacle of this approach is that it does not perform well on very large networks. Using current computer hardware and recent MINLP solvers, networks with up to 1,000 nodes can routinely be solved with this approach [25]. For networks larger than this, it turned out to be too slow. Hence alternative solution techniques are sought.

For the special case of purely passive networks (without active elements) and a uniform height level, Maugis [21] (gas networks) and, independently, Collins et al. [5] (water networks) presented an approach, that we will present below. The algorithmic approach of Maugis and Collins et al. is based on a relaxation of the bounds on the flow and on the pressure variables by introducing slack variables. The sum of these slacks is then minimized via the objective function. This results in a convex optimization problem, hence any local optimal solution (found by a local NLP solver) is already a global solution. The values of the pressure variables can then by moved without altering the values of the flow variables. If it is possible to move the pressure values inside the (previously relaxed) bounds, then a feasible solution for the flow problem is found. If not, then this is a certificate for the infeasibility of the problem. This approach was used as starting point for more refined computations in further studies of gas and water network problems, see for instance, Sherali and Smith [26], De Wolf et al. [7–11], or Bobonneau et al. [2].

In the following we will present a new pivot strategy for the existing combinatorial algorithm of Osiadacz [24] to solve Maugis' optimization problem. Using this strategy, we are able to prove a convergence rate result for the algorithms. One of them is based on an iterative change of the flow values, the other on an iterative alteration of the pressure values. Since our work applies also to other types of nonlinear network flow problem, such as water networks or electricity networks, we do no speak of a node pressure, but more generically, of a node potential (level).

The nonlinear flow model – to be presented in detail below – consists of three types of constraints: first, trivial bounds on the flow and potential variables, second, linear flow conservation constraints, and third, nonlinear potential-flow coupling constraints that define a relationship between the flow value on an arc and the potential levels at the two end nodes. The approach of Maugis is a relaxation of the first constraint type. It is natural to ask whether it is possible to algorithmically exploit a relaxation based on the second or the third type of constraints. This analysis is content of the remaining part of this article.

We relax the linear flow conservation constraints. To this end, we introduce also slack variables that inherit the residual value that the incoming and outgoing flows in a node deviate from equality. These slack variables are coupled to the potential differences by complementarity constraints. We extend the nonlinear optimization problem of Maugis to this case. Also this problem turns out to be convex, so that a local optimal solution is already a global one. Instead of general purpose NLP solvers we present a combinatorial algorithm for its solution. We demonstrate that the value of the slack variables yield a certificate of infeasibility.

Finally, we relax the potential-flow coupling constraints. This also gives a certificate for an instance being infeasible. However, the solution is not as unique as in the other two cases.

Using larger networks we derive computational results to compare the running times. Starting with the full MINLP formulation and spatial branching, we continue with NLP solvers for all three relaxations. Our strategy of using an NLP solver as subproblem solver in an MINLP problem is similar to a solution process described by Gentilini et al. [18], which they apply for the solution of a nonlinear variant of the traveling salesman problem.

Finally, we will extend Maugis' work, so that also active elements and inhomogeneous height levels can be treated. Here we give a certificate of infeasibility using the solution to the dual of the flow problem. The dual variables can also be seen as flow variables. Their values in an optimal solution indicate where the potential loss is too big. With their help it is possible to identify a pair of nodes that are the source of infeasibility. This result can be seen as an equivalent to the classical max-flow min-cut theorem of Ford and Fulkerson [14] for linear flows (in particular, flows without potentials), where the cut indicates the bottleneck of the network that limits a further growth of the flow.

2 Mathematical Background

In order to obtain proven global optimal solutions we apply linear and nonlinear mixed-integer programming techniques, which we briefly introduce here.

2.1 Global Mixed-Integer Nonlinear Programming

We formulate the nonlinear network flow problem in an active (i.e., controllable) network as mixed-integer nonlinear program (MINLP). Solving optimization problems from this class is theoretically intractable and also known to be computationally difficult in general. By "solving" we mean to compute a feasible solution for a given instance of the problem together with a computational proof of its optimality. Therefor we apply the general framework of a branch-and-bound approach, where the bounds are obtained from relaxations of the original model. To this end, we relax the MINLP first to a mixed-integer linear program (MILP) and then further to a linear program (LP), which is solved efficiently using the simplex algorithm. The so obtained solution value defines a (lower) bound on the optimal value of the original MINLP problem. In case this solution is MINLP feasible, it would be a proven global optimal MINLP solution. However, this rarely happens in practice. Hence we either add cutting planes to strengthen the relaxation, or we decide to branch on a variable. As an example, consider the nonlinear potential loss constraint (3.1), c.f. Fügenschuh et al. [15]. In the LP relaxation this function is replaced by a polyhedral (linear) outer approximation, which is iteratively refined during the branch-and-bound process by branching on variables (spatial branching), see Figure 2.1. For more details on cutting planes and branch-and-bound for MILP we refer to Nemhauser and Wolsey [22], and for an application of this framework to global mixed-integer nonlinear programming to Smith and Pantelides [27], and Tawarmalani and Sahinidis [28, 29]. Information on the framework MINLP framework SCIP which we apply is given by Achterberg [1], and in particular on nonlinear aspects of SCIP in Berthold, Heinz, and Vigerske [3].

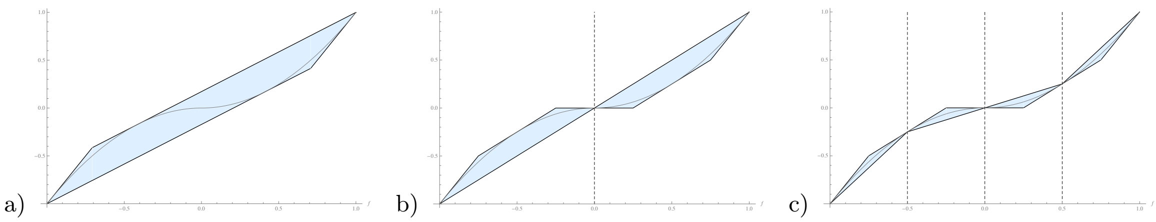


Fig. 2.1. a) Polyhedral outer approximation of $q_e \mapsto \alpha_e q_e |q_e|$, b) initial spatial branching on zero, c) further spatial branching.

2.2 Nonlinear Programming

In addition to the simplex algorithm for linear programs we use nonlinear solvers on nodes of the branch-and-bound tree. As soon as all binary decisions and the flow directions are fixed, the remaining problem at this node is (equivalent to) a convex nonlinear problem (see Collins et al. [5]). To compute optimal solutions for these subproblems we apply the solver IPOPT from Wächter and Biegler [30]. It applies a primal-dual interior point (or barrier) method with a filter line-search method. One of the central underlying method in nonlinear programming, which is part of in IPOPT and which we also apply directly in our solution approach, are the Karush-Kuhn-Tucker (KKT) conditions. Under certain additional assumptions they provide necessary conditions for a (local) optimum. For a nonlinear optimization problem of the form $\min\{f(x) : g_i(x) \leq 0, h_j(x) = 0, x \in \mathbb{R}^n\}$, where f is the objective function, $g_i (i = 1, \ldots, m)$ are continuously differentiable inequality constraint functions and $h_j (j = 1, \ldots, \ell)$ are continuously differentiable equality constraint functions, the KKT system reads as follows

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) + \sum_{j=1}^{\ell} \mu_j \nabla h_j(x^*) = 0, \tag{2.1}$$

$$g_i(x^*) \leq 0, \forall\, i = 1, \ldots, m, \tag{2.2}$$

$$h_j(x^*) = 0, \forall\, j = 1, \ldots, \ell, \tag{2.3}$$

$$\lambda_i \geq 0, \forall\, i = 1, \ldots, m, \tag{2.4}$$

$$\lambda_i g_i(x^*) = 0, \forall\, i = 1, \ldots, m, \tag{2.5}$$

where x^* is a local minimum, and $\lambda_i (i = 1, \ldots, m), \mu_j (j = 1, \ldots, \ell)$ are constants (called KKT multipliers). In the special case of $m = 0$, i.e., no inequality constraints exist, the KKT multipliers are also called Lagrange multipliers. For more details we refer to Conn, Gould, and Toint [6].

3 Physical and Technical Background of Transmission Networks

We give mathematical descriptions for active and passive elements that are the basic building blocks of the transmission networks we study (see also [16]).

3.1 Pipes

The majority of the edges in a transmission network are passive pipes. In a network with node potentials the amount of flow over an edge is determined by the actual node potential values at both ends. Depending on the physical properties of the flow different functional relationships are described in the literature to approximatively determine the flow value. The fundamental equation we assume for an edge $e = (v, w)$ is

$$\alpha_e q_e |q_e|^{k_e} = \pi_v - \gamma_e \pi_w. \tag{3.1}$$

Here α_e, k_e, γ_e are constants that subsume all physical properties of the edge, the flow, and the interactions of the flow with the edge (c.f. Weymouth [31] for gas and Hazen-Williams [19] for water). The constant γ_e in particular represents the height difference between nodes v and w. If some pipelines $e_1, \ldots, e_n$ form a cycle, it is assumed that $\gamma_{e_1} \cdot \ldots \cdot \gamma_{e_n} = 1$. If $e = (v, w)$ is an edge and $e' = (w, v)$ is its anti-parallel counterpart, then we assume that the constants γ_e are such that $\gamma_e = \gamma_{e'}^{-1}$. A generalized Weymouth equation with this properties can be found in [25]. Although each edge e in principle might have a different value for k_e it is natural to assume that all edges have the same constant. The variable $q_e \in \mathbb{R}$ represents the flow, where a positive value is a flow from v to w, and a negative value is a flow in the opposite direction from w to v. The variables π_v, π_w are the node potential values.

3.2 Valves

A valve is installed in the network to separate or join two independent pipes. They allow for a discrete decision, either being open or close. The spatial dimension of a valve is assumed to be small in comparison to the pipes. Hence in our model the node potential values are identified when the valve is open. If the valve is closed then they are decoupled. Mathematically a valve is an edge $e = (v, w)$ with the following description:

$$x_e = 1 \Rightarrow \;\pi_v - \pi_w = 0, \tag{3.2a}$$

$$x_e = 0 \Rightarrow \;q_e = 0, \tag{3.2b}$$

where $x_e \in \{0, 1\}$ is a binary decision variable.

3.3 Increasing the Node Potential

In transmission networks it is necessary at certain locations to increase the node potential value. For example, in gas networks the potential is too low after a transport distance of 100-150km. Here gas turbines are used as compressors. For the mathematical description of such active network elements, various models exist in the literature. We follow the approach of De Wolf and Smeers [12], and make use of the following formulation for a pipe $e = (v, w)$ with a compressor:

$$\alpha_e q_e |q_e|^{k_e} \geq \pi_v - \pi_w, \tag{3.3}$$

which allows a flow larger than the one corresponding to the potential decrease in the pipe. We rewrite this inequality as equality by introducing a weighted slack variable y_e:

$$\alpha_e q_e |q_e|^{k_e} + \beta_e y_e = \pi_v - \pi_w. \tag{3.4}$$

Note that the flow can only go in positive direction through a compressor, hence the lower bound needs to be set accordingly, i.e., $q_e \geq 0$.

3.4 Reducing the Node Potential

It can be necessary to reduce the node potential along an edge $e = (v, w)$ in the network, for example, to protect parts of the network from high potentials. In gas networks, for instance, these are potential regulation stations that reduce the gas pressure. A pipe with a pressure regulator $e = (v, w)$ is inverse to a pipe with a compressor. Hence we need to turn the sense of the inequality (3.3) around:

$$\alpha_e q_e |q_e|^{k_e} \leq \pi_v - \pi_w, \tag{3.5}$$

in order to decrease the potential in w more than the flow and the input potential (pressure) would actually require. After introducing weighting slack variables y_e equation (3.5) appears similar to equation (3.4). (The only difference between a compressor and a regulator is either the sign of β or the bounds on y_e.) Note that the flow direction through a pressure regulator is also fixed by setting the lower bound to zero, i.e., $q_e \geq 0$.

4 Flow Optimization of Transmission Networks

In the following we describe a mixed-integer nonlinear model for the nonlinear flow problem in a transmission network. In [16] we presented an extension of this model that includes also pipeline extension decisions, in case that the given network has not enough capacity for a given amount of flow at the entry and exit nodes.

4.1 The Model

We use the following notation for sets. A transmission network is modeled by a directed graph $G = (V, E)$ where V denotes the set of nodes and $E \subset V \times V$ the set of arcs. Each active or passive network element that connects nodes v and w is represented by an arc (v, w). The set $E = E_a \cup E_p$ is the disjoint union of the set of active elements E_a and the set of passive elements E_p. For a given subset of nodes $S \subseteq V$ we denote by $\delta_G(S)$ the set of all arcs in G that have exactly one node $v \in S$ either as start or end node, i.e., $\delta_G(S) := \{e = (v, w) \in E : v \in S, w \notin S\} \cup \{e = (v, w) \in E : v \notin S, w \in S\}$. We denote by $\Gamma_G(S)$ the set of all nodes in $V \backslash S$ that are neighbors of nodes in S, i.e., $\Gamma_G(S) := \{w \in V \backslash S : \exists v \in S, (v, w) \in E\} \cup \{w \in V \backslash S : \exists v \in S, (w, v) \in E\}$. Since we usually operate on one single graph at a time, we skip the index and simply write $\delta(S)$ and $\Gamma(S)$. For a set $S = \{v\}$ we also write $\delta(v)$ and $\Gamma(v)$. We also use the abbreviations $\delta^+_G(v) := \{e \in E \mid \exists w \in V, e = (v, w)\}$ for the set of outgoing arcs and $\delta^-_G(v) := \{e \in E \mid \exists w \in V, e = (w, v)\}$ for the set of ingoing arcs w.r.t. node v, and generally omit the subscript G.

We assume the following data to be given as parameters. For each node $v \in V$ we have lower and upper bounds on the node potential, $\underline{\pi}_v, \overline{\pi}_v \in \mathbb{R}$ with $\underline{\pi}_v \leq \overline{\pi}_v$. For each node $v \in V$ the value $s \in \mathbb{R}$ denotes the amount of flow that is either led into the network (for $s_v > 0$), or taken out of the network (for $s_v < 0$). A node with $s_v > 0$ is also called source or entry node, and nodes with $s_v < 0$ are sinks or exit nodes. All other nodes with $s_v = 0$ are inner or transmission nodes. Vector v is also called *nomination.* In order not to pose a problem that is trivially infeasible, only those nominations are allowed that have equal entry and exit flows, that is,

$$\sum_{v \in V} s_v = 0. \tag{4.1}$$

Such nominations are said to be *balanced.* For each arc $e = (v, w) \in E$ we have a transmission coefficient $\alpha_e \in \mathbb{R}_+ \backslash \{0\}$, bounds on the range coefficient $\underline{y}_e, \overline{y}_e \in \mathbb{R}$ with $\underline{y}_e \leq \overline{y}_e$, and a scaling factor β_e for the range coefficient.

Let us introduce the following variables. The flow on arc $e \in E$ is denoted by $q_e \in \mathbb{R}$, where a positive value means the flow is heading in the same direction as the arc, and a negative value indicates the opposite direction. The potential value of a vertex $v \in V$ is given by $\pi_v \in \mathbb{R}$. For example, in a gas transmission network this variable refers to the potential in this node. The variable $y_e \in \mathbb{Z}$ is a multiplier for the additive component of the potential loss term. For passive pipelines this variable is fixed to zero, whereas for active elements it defines the operating range.

The question of the existence of a feasible solution vector $(x_e, y_e, q_e, \pi_v)_{e \in E, v \in V}$ for the following non-linear non-convex mixed-integer model is called *nonlinear transmission flow feasibility problem* (or *flow problem*, for short):

$$x_e = 1 \Rightarrow \alpha_e q_e |q_e|^{k_e} + \beta_e y_e - (\pi_v - \gamma_e \pi_w) = 0 \qquad \forall\, e = (v, w) \in E_a, \tag{4.2a}$$

$$\alpha_e q_e |q_e|^{k_e} - (\pi_v - \gamma_e \pi_w) = 0 \qquad \forall\, e = (v, w) \in E_p, \tag{4.2b}$$

$$x_e = 1 \Rightarrow y_e \leq \overline{y}_e \qquad \forall\, e \in E_a, \tag{4.2c}$$

$$x_e = 1 \Rightarrow y_e \geq \underline{y}_e \qquad \forall\, e \in E_a, \tag{4.2d}$$

$$x_e = 0 \Rightarrow q_e = 0 \qquad \forall\, e \in E_a, \tag{4.2e}$$

$$x_e = 0 \Rightarrow y_e = 0 \qquad \forall\, e \in E_a, \tag{4.2f}$$

$$\sum_{e \in \delta^+(v)} q_e - \sum_{e \in \delta^-(v)} q_e = s_v \qquad \forall\, v \in V, \tag{4.2g}$$

$$\pi_v \leq \overline{\pi}_v \quad \forall\, v \in V, \tag{4.2h}$$

$$\pi_v \geq \underline{\pi}_v \quad \forall\, v \in V, \tag{4.2i}$$

$$q_e \leq \overline{q}_e \quad \forall\, e \in E, \tag{4.2j}$$

$$q_e \geq \underline{q}_e \quad \forall\, e \in E, \tag{4.2k}$$

$$x_e \leq 1 \quad \forall\, e \in E_a, \tag{4.2l}$$

$$x_e \geq 0 \quad \forall\, e \in E_a, \tag{4.2m}$$

$$q_e \in \mathbb{R} \quad \forall\, e \in E, \tag{4.2n}$$

$$\pi_v \in \mathbb{R} \quad \forall\, v \in V, \tag{4.2o}$$

$$y_e \in \mathbb{Z} \quad \forall\, e \in E_a, \tag{4.2p}$$

$$x_e \in \mathbb{Z} \quad \forall\, e \in E_a. \tag{4.2q}$$

The indicator constraints (4.2a) are switching on only those potential-flow coupling constraints for active arcs that are actually open. For passive arcs we have the pressur-flow coupling constraints (4.2b). The indicator constraints (4.2c) and (4.2d) enable the selection of an operating mode if an active element is open. The indicator constraints (4.2e) forbid flow on those active arcs that are actually closed, that is, they are switched off by a closed valve. The indicator constraints (4.2f) switch off the control valve or compressor, when this active element is closed. The node flow conservation constraints (also called Kirchhoff's constraints) are defined in (4.2g). Constraints (4.2h) – (4.2m) define the trivial bounds on the variables, and constraints (4.2n) – (4.2q) specify the continuous or discrete range of the variables.

For a given nomination s, the flow problem (4.2) is to find a setting of the active elements and flow and potential values for the transmission of the specific flow s in the transmission network G. Otherwise, if this transport is not possible for any setting of the active elements, the nomination is infeasible.

After branching on x and y we obtain the *leaf problem* as a nonlinear subproblem within the branching tree: We define $E'_a := \{e \in E_a : x_e = 1\}$ and $E' := E'_a \cup E_p$ and $\tilde{\beta}_e := \beta_e y_e$. We assume $\tilde{\beta}_e = 0$ for $e \in E_p$.

$$\alpha_e q_e |q_e|^{k_e} + \tilde{\beta}_e - (\pi_v - \gamma_e \pi_w) = 0 \qquad \forall\, e = (v, w) \in E', \tag{4.3a}$$

$$\sum_{e \in \delta^+(v)} q_e - \sum_{e \in \delta^-(v)} q_e = s_v \quad \forall\, v \in V, \tag{4.3b}$$

$$\pi_v \leq \overline{\pi}_v \quad \forall\, v \in V, \tag{4.3c}$$

$$\pi_v \geq \underline{\pi}_v \quad \forall\, v \in V, \tag{4.3d}$$

$$q_e \leq \overline{q}_e \quad \forall\, e \in E', \tag{4.3e}$$

$$q_e \geq \underline{q}_e \quad \forall\, e \in E', \tag{4.3f}$$

$$\pi_v \in \mathbb{R} \quad \forall\, v \in V, \tag{4.3g}$$

$$q_e \in \mathbb{R} \quad \forall\, e \in E'. \tag{4.3h}$$

5 Domain Relaxations

Let us consider the following *domain relaxation* of a leaf problem. We introduce a slack variable $\Delta_v \in \mathbb{R}_+$ for the potential of node $v \in V$ and another slack variable $\Delta_e \in \mathbb{R}_+$ for the flow of arc $e \in E'$, and obtain the following nonlinear optimization problem:

$$\min \sum_{v \in V} \Delta_v + \sum_{e \in E'} \Delta_e \quad \text{s. t.}$$

$$\alpha_e q_e |q_e|^{k_e} + \tilde{\beta}_e - (\pi_v - \gamma_e \pi_w) = 0 \qquad \forall\, e = (v,w) \in E', \tag{5.1a}$$

$$\sum_{e\in\delta^+(v)} q_e - \sum_{e\in\delta^-(v)} q_e = s_v \qquad \forall\, v \in V, \tag{5.1b}$$

$$\pi_v - \Delta_v \le \overline{\pi}_v \qquad \forall\, v \in V, \tag{5.1c}$$

$$\pi_v + \Delta_v \ge \underline{\pi}_v \qquad \forall\, v \in V, \tag{5.1d}$$

$$q_e - \Delta_e \le \overline{q}_e \qquad \forall\, e \in E', \tag{5.1e}$$

$$q_e + \Delta_e \ge \underline{q}_e \qquad \forall\, e \in E', \tag{5.1f}$$

$$\pi_v \in \mathbb{R} \qquad \forall\, v \in V, \tag{5.1g}$$

$$q_e \in \mathbb{R} \qquad \forall\, e \in E', \tag{5.1h}$$

$$\Delta_v \in \mathbb{R}_+ \quad \forall\, v \in V. \tag{5.1i}$$

$$\Delta_e \in \mathbb{R}_+ \quad \forall\, e \in E', \tag{5.1j}$$

Note that the optimal objective value of (5.1) equals zero if and only if the leaf problem (4.3) has a solution. In this case its solution is feasible to the leaf problem. In the following we show how to solve this relaxed leaf problem (5.1) to global optimality.

5.1 Computing Solutions based on Convex NLPs

The existence of a primal solution to (5.1) is shown by Collins et al. [5]. In the following, we review their method. Note that it only works for constant heights, i.e., $\gamma_e = 1$ for all $e \in E'$. In the next section we will show how the case of inhomogeneous heights can be treated as an aftermath.

Collins et al. considered the convex non-linear optimization problem

$$\begin{aligned} \min \sum_{e\in E'} \int_{q_e^0}^{q_e} \Phi_e(t)\ \mathrm{d}t \quad \text{s.t.} & \\ \sum_{e\in\delta^-(v)} q_e - \sum_{e\in\delta^+(v)} q_e = -s_v & \qquad \forall\, v \in V, \\ q_e \in \mathbb{R} & \qquad \forall\, e \in E', \end{aligned} \tag{5.2}$$

where $\Phi(\cdot)$ is a continuous strictly monotone function and q_e^0 is a root of $\Phi_e(\cdot)$ which implies that the objective is convex. In the context of our study we set $\Phi_e(q_e) := \alpha_e q_e |q_e|^{k_e} + \tilde{\beta}_e$. From the KKT conditions we obtain primal (local) optimal values q^* and dual values μ^* such that

$$\mu_v^* - \mu_w^* = \Phi_e(q_e^*).$$

Hence with the setting $\pi^* := \mu^*$ we obtain (together with q^*) a primal feasible solution of (5.1) with $\gamma_e = 1$ (for all $e \in E'$).

An alternative proof given by Collins et al. to construct a solution for (5.1) (with $\gamma_e = 1$) is to consider the following non-linear convex program:

$$\begin{aligned} \min \sum_{e=(v,w)\in E'} \int_{\Delta_e^0}^{\pi_v - \pi_w} \Phi_e^{-1}(t)\mathrm{d}t - \sum_{v\in V} \int_0^{\pi_v} s_v \mathrm{d}t \quad \text{s.t.} & \\ \pi \in \mathbb{R}^V. & \end{aligned} \tag{5.3}$$

Here Δ_e^0 is a root of Φ_e^{-1}. For an optimal solution π^* we define q^* by

$$q_e^* := \Phi_e^{-1}(\pi_v^* - \pi_w^*) \quad \Leftrightarrow \quad \Phi_e(q_e^*) = \pi_v^* - \pi_w^*.$$

By the KKT conditions we obtain that the following flow conservation constraints are fulfilled

$$\sum_{e\in\delta^+(v)} q_e^* - \sum_{e\in\delta^-(v)} q_e^* - s_v = 0 \quad \forall\, v \in V.$$

5.2 Characterizing the Feasible Region

Theorem 1. *There exists a vector* $\tilde{q} \in \mathbb{R}^{E'}$*, a vector* $\tilde{\pi} \in \mathbb{R}^V$ *and a vector* $\theta \in \mathbb{R}^V \backslash \{0\}$*, such that for any feasible solution* (q^*, π^*) *of problem* (5.1) *there exists a* $t \in \mathbb{R}$ *with* $\pi^* = \tilde{\pi} + t\theta$ *and* $q^* = \tilde{q}$*.*

By this theorem, the feasible region of problem (5.1) is a convex space: The feasible flow q is unique, while the feasible potential values π are on a straight line.
Proof: First we prove that the primal solution flow of (5.1) is unique. We assume the existence of two different solutions (π', q') and (π'', q'') of (5.1) with $q' \neq q''$. The difference $q' - q''$ is a network flow in (V, E') consisting of circulations only. Let us consider a circuit C such that w.l.o.g. $q'_e - q''_e > 0$ for all arcs $e \in E'(C)$ (we might have to change the orientation of an arc in case the difference is negative). As abbreviation we set $\Phi_e(q) := \alpha_e q |q|^{k_e} + \tilde{\beta}_e$. We use the strong monotonicity of Φ_e and equation (5.1a) to obtain for all $e = (v, w) \in E'(C)$:

$$q'_e > q''_e \Leftrightarrow \Phi_e(q'_e) > \Phi_e(q''_e) \Leftrightarrow \pi'_v - \gamma_e \pi'_w > \pi''_v - \gamma_e \pi''_w.$$

Let the nodes of the circuit C be ordered such that $V(C) = \{v_1, \ldots, v_\ell\}$ and (as abbreviation) $v_{\ell+1} := v_1$, and the arcs be ordered such that $e_i = (v_i, v_{i+1})$ holds. By a telescope sum argument we derive the contradiction

$$\sum_{i=1}^{\ell} \left(\prod_{j=1}^{i-1} \gamma_{e_j} \right) \left(\pi'_{v_i} - \gamma_{e_i} \pi'_{v_{i+1}} \right) > \sum_{i=1}^{\ell} \left(\prod_{j=1}^{i-1} \gamma_{e_j} \right) \left(\pi''_{v_i} - \gamma_{e_i} \pi''_{v_{i+1}} \right)$$

$$\Leftrightarrow \pi'_{v_1} \left(1 - \prod_{e \in E'(C)} \gamma_e \right) > \pi''_{v_1} \left(1 - \prod_{e \in E'(C)} \gamma_e \right).$$

Note that $\prod_{e \in E'(C)} \gamma_e = 1$ (cf. Section 3.1), hence we have $\pi'_{v_1} \cdot 0 > \pi''_{v_1} \cdot 0$, which is a contradiction. Hence the assumption, that the flow is not unique, was wrong. So the solution flow $q^* = \tilde{q} = q' = q''$ of (5.1) is unique.

Now we prove that the feasible solution potentials of (5.1) form a straight line. Therefor let (π', q) and (π'', q) be two different feasible solutions of (5.1). We select any node $r \in V$ as root node. Let $w \in V$ be any other node in V. Consider a r-w-path P with nodes $\{r = v_1, \ldots, v_k = w\}$ and arcs $\{e_1, \ldots, e_{k-1}\}$. We obtain the equality

$$\pi'_r - \pi'_w \prod_{j=1}^{k-1} \gamma_{e_j} = \sum_{j=1}^{k-1} \left(\prod_{i=1}^{j-1} \gamma_{e_i} \right) \Phi_{e_j}(q_{e_j}) = \pi''_r - \pi''_w \prod_{j=1}^{k-1} \gamma_{e_j}.$$

This is equivalent to

$$\pi'_r - \pi''_r = (\pi'_w - \pi''_w) \prod_{e \in E'(P)} \gamma_e.$$

We define $\theta_w := (\prod_{e \in E'(P)} \gamma_e)^{-1} \neq 0$. This setting is well-defined, i.e., independent from the actual path P from r to w: Let P' be a different r-w-path. Consider the cycle C from r to w on path P, and back from w to r on path P' in reverse order. Denote the reverse path of P' by Q'. For this cycle we have that $1 = \prod_{e \in E'(C)} \gamma_e = \prod_{e \in E'(P)} \gamma_e \cdot \prod_{e \in E'(Q')} \gamma_e = \prod_{e \in E'(P)} \gamma_e \cdot (\prod_{e \in E'(P')} \gamma_e)^{-1}$, hence $\prod_{e \in E'(P)} \gamma_e = \prod_{e \in E'(P')} \gamma_e$.

We set $t := \pi''_r - \pi_r$. Then the solution π'' can be expressed as $\pi''_w = \pi'_w + t\theta_w$ for all $w \in V$. This proves the theorem. □

5.3 Height and Optimality

Given a feasible solution (q^*, π^*) for (5.1) with $\gamma_e = 1$ (for all $e \in E'$), we show how to obtain first a feasible solution for (5.1) with arbitrary values of γ, and then an optimal solution for (5.1).

We select any node $r \in V$ as root node. Let w be any other node in V. Let P be a r-w-path with nodes $r = v_1, \ldots, v_\ell = w$ and arcs $e_1, \ldots, e_{\ell-1}$. We set π'_w by

$$\pi'_w := \prod_{i=1}^{\ell-1} \gamma_{e_i}^{-1} \left(\pi^*_r - \left(\sum_{j=1}^{\ell-1} \Phi_{e_j}(q^*_{e_j}) \prod_{i=1}^{j-1} \gamma_{e_i} \right) \right)$$

for each node $w \in V$. Then (q^*, π') is a feasible solution for the relaxed leaf problem (5.1), but not necessarily an optimal one.

In order to obtain an optimal solution for (5.1) we simply fix the variables q to q^* in (5.1), and solve the remaining LP problem in π, Δ_e, Δ_v.

Figure 5.1 shows the potential values for a test network having 34 nodes. The node potential bounds are shown as straight lines (lower bound of 500 and upper bound of 6000). Solutions for four different leafs in the branch-and-bound tree are shown with four different colors: Each dot represents the node potential value at the respective node. For three of the four problems it was not possible to move all potential values inside the bounds, hence these solutions are infeasible. Only for the solution corresponding to the green colored dots, all values are inside the bounds, and this solution is feasible.

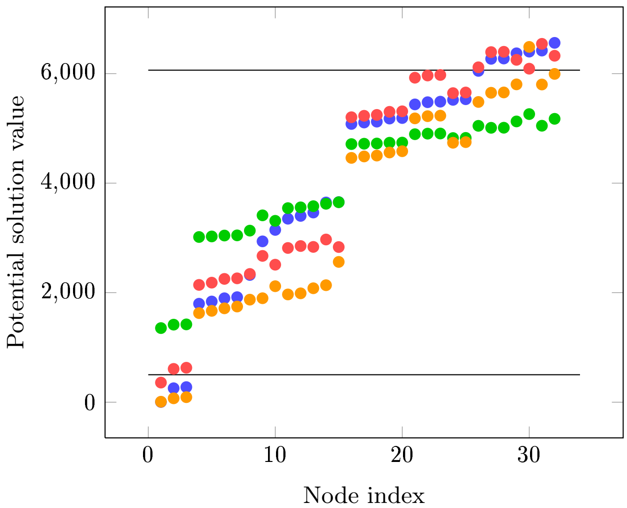


Fig. 5.1. Node potential values for four different leaf problems. The node potential values are feasible within the marked bounds, other values are infeasible.

5.4 The Potential Adjustment Algorithm

We give a combinatorial algorithm to compute a feasible solution for (4.3) or detect infeasibility. Indeed the algorithm computes a feasible solution (q^*, π^*) to the relaxed leaf problem (5.1) with $\gamma_e = 1$ for all $e \in E'$, which might not be optimal. As shown in Section 5.3 we can modify the feasible potential π in an aftermath, such that the solution (q^*, π^*) is either feasible to the leaf problem (4.3) (for arbitrary values of γ_e) or we detect the infeasibility of the leaf problem.

The outline of the algorithm is as follows. We set $\gamma_e := 1$ for all $e \in E'$. We start with $\pi_v = 0$ for all $v \in V$ (or any other potential values). This uniquely determines the value q_e for all $e \in E'$ using equation (5.1a). In each subsequent iteration of the algorithm, the node potential values π are either increased or decreased. To this end, the potential value π_v for node v is updated by computing the unique solution to the nonlinear equation system

$$\sum_{e \in \delta^+(v)} q_e - \sum_{e \in \delta^-(v)} q_e = s_v, \tag{5.4a}$$

$$\Phi_e(q_e) = \pi_v - \tilde{\pi}_w, \quad \forall e = (v, w) \in E' \tag{5.4b}$$

$$\Phi_e(q_e) = \tilde{\pi}_w - \pi_v, \quad \forall e = (w, v) \in E'. \tag{5.4c}$$

In this system, the values for $\tilde{\pi}_w$ are fixed, and the unknown variables are π_v and q_e for all arcs e that are incident to v. Hence the system has $\deg(v) + 1$ many variables and constraints. To see that the solution is unique, we transform (5.4b) and (5.4c) by the inverse Φ_e^{-1} (remember that Φ is bijective), and obtain the equations $q_e = \Phi^{-1}(\pi_v - \tilde{\pi}_w)$ and $q_e = \Phi^{-1}(\tilde{\pi}_w - \pi_v)$. Note that Φ is a strictly monotone growing function, hence Φ^{-1} is also strictly monotone growing. Hence $\pi_v \mapsto \Phi^{-1}(\pi_v - \tilde{\pi}_w)$ and $\pi_v \mapsto -\Phi^{-1}(\tilde{\pi}_w - \pi_v)$ are also strictly monotone growing functions. Hence their sum is also a strictly monotone growing function, and therefore equation (5.4a) offers a unique solution for π_v:

$$\sum_{(w,v)\in\delta^+(v)} \Phi^{-1}(\tilde{\pi}_w - \pi_v) - \sum_{(v,w)\in\delta^-(v)} \Phi^{-1}(\pi_v - \tilde{\pi}_w) = s_v. \tag{5.5}$$

With this so-computed value for π_v we can uniquely compute values for each q_e using (5.4b) and (5.4c), respectively.

Hence the potential-flow coupling equation (5.1a) is fulfilled throughout the algorithm, whereas the flow conservation equation (5.1b) might not hold during the execution of the algorithm. We define the following error $\varepsilon(v)$ that measures the deviation of the left-hand and right-hand side of this equation:

$$\varepsilon(v) := s_v + \sum_{e\in\delta^-(v)} q_e - \sum_{e\in\delta^+(v)} q_e.$$

Note that $\sum_{v\in V} \varepsilon(v) = 0$, because

$$\begin{aligned}\sum_{v\in V} \varepsilon(v) &= \sum_{v\in V}\left(s_v + \sum_{e\in\delta^-(v)} q_e - \sum_{e\in\delta^+(v)} q_e\right)\\ &= \sum_{v\in V} s_v + \sum_{v\in V}\left(\sum_{e\in\delta^-(v)} q_e - \sum_{e\in\delta^+(v)} q_e\right) = 0 + 0 = 0.\end{aligned}$$

From this it follows, that it is only necessary to consider the error of nodes with positive error. If all errors are $\varepsilon(v) \leq 0$ it follows that $\varepsilon(v) = 0$.

The total error $\mathcal{E}$ is defined as the sum of the absolute values for all node errors, i.e., $\mathcal{E} := \sum_{v\in V} |\varepsilon(v)| \geq 0$. The goal is to decrease the total error in each iteration, until it reaches zero, see Algorithm 1 for the details. Finally, a feasible flow is found. In a final step the node potential values are determined as described in Section 5.3, and it is then decided, whether (4.3) has a feasible solution or not.

Algorithm 1: PressureAdjustment(G', π)

1 **while** $\mathcal{E} > 0$ **do**
  $i \leftarrow 1$
  Let $v_1 \in \arg\max\{\varepsilon(v) > 0 : v \in V\}$.
  Set $S := \{v_1\}$.
  Update π_{v_1} by solving equation system (5.4).
2   **while** $S \neq V$ **do**
    $i \leftarrow i + 1$
    Let $v_i \in \Gamma(S)$.
    if $\varepsilon(v_i) > 0$ **then**
      Update π_{v_i} by solving equation system (5.4).
    $S \leftarrow S \cup \{v_i\}$.

By $\langle j, i\rangle$ we denote an iteration counter for Algorithm 1, indicating that the algorithm is in the j-th outer loop 1 and in the i-th inner loop 2 (for $i \leq n$), when the node potentials for the nodes $v_1, \ldots, v_{i-1}$ were already updated, and the node potential of node v_i is not yet updated. By $\varepsilon_{j,i}(v)$ we denote the error $\varepsilon(v)$ and by $\mathcal{E}_{j,i}$ we denote the total error $\mathcal{E}$ in iteration $\langle j, i\rangle$. Let $\mathcal{E}_j$ denote the error $\mathcal{E}$ after j iterations of the outer loop 1, an let $\mathcal{E}_{j,i}$ denote the error $\mathcal{E}$ in iteration $\langle j, i\rangle$.

Lemma 1. *The error in Algorithm 1 is non-increasing, i.e., $\mathcal{E}_{j,i+1} \leq \mathcal{E}_{j,i}$. Hence $\mathcal{E}_{j+1} \leq \mathcal{E}_j$.*

Proof: Let $v_i \in V$ be the vertex which is processed in iteration $\langle j, i\rangle$. We assume w.l.o.g. that every edge incident to v_i is directed away from v_i. Let π, q belong to iteration $\langle j, i\rangle$ and π', q' belong to iteration

$\langle j, i+1\rangle$. As abbreviation we identify iteration $\langle j, n+1\rangle$ with iteration $\langle j+1, 1\rangle$. By definition of the update of the potential at v_i we have that a) $\pi'_w = \pi_w$ at every node except v_i, b) $\pi'_{v_i} \neq \pi_{v_i}$, c) $q'_e \neq q_e$ for all edges e incident with v_i, and d) $q'_e = q_e$ for every other edge (i.e., all edges e not incident to v_i). Moreover, $\varepsilon_{j,i}(v_i) = \sum_{e\in\delta(v_i)}(q'_e - q_e) \neq 0$ while $\varepsilon_{j,i+1}(v_i) = 0$. The new error at some neighbor w of v_i is (as we only have outgoing edges)

$$\begin{aligned}\varepsilon_{j,i+1}(w) &= s_w + \sum_{e\in\delta^-(w)} q'_e - \sum_{e\in\delta^+(w)} q'_e \\ &= s_w + \sum_{e\in\delta^-(w)} q_e - \sum_{e\in\delta^+(w)} q_e - (q'_{v_i,w} - q_{v_i,w}) \\ &= \varepsilon_{j,i}(w) - \underbrace{(q'_{v_i,w} - q_{v_i,w})}_{\geq 0}.\end{aligned}$$

Thus

$$\begin{aligned}\mathcal{E}_{j,i+1} &= \sum_{w\in V} |\varepsilon_{j,i+1}(w)| \\ &= \sum_{w\in V - v_i - \Gamma(v_i)} |\varepsilon_{j,i}(w)| + \sum_{w\in\Gamma(v_i)} |\varepsilon_{j,i+1}(w)| + \underbrace{\varepsilon_{j,i+1}(v_i)}_{=0} \\ &\leq \sum_{w\in V - v_i - \Gamma(v_i)} |\varepsilon_{j,i}(w)| + \sum_{w\in\Gamma(v_i)} (|\varepsilon_{j,i}(w)| + q'_{v_i,w} - q_{v_i,w}) \\ &\leq \sum_{w\in V - v_i} |\varepsilon_{j,i}(w)| + \sum_{w\in\Gamma(v_i)} (q'_{v_i,w} - q_{v_i,w}) \\ &\leq \sum_{w\in V - v_i} |\varepsilon_{j,i}(w)| + \underbrace{\sum_{e\in\delta(v_i)} (q'_e - q_e)}_{=\varepsilon_{j,i}(v_i)} \\ &= \sum_{w\in V} |\varepsilon_{j,i}(w)| = \mathcal{E}_{j,i}.\end{aligned}$$

So the total error cannot increase. □

Lemma 2. *Let $b, c \in \mathbb{R}$ and $k \in \mathbb{R}_+$ with $k \geq 1$. Let $f : \mathbb{R} \to \mathbb{R}$ be a function defined by $f(x) := c \cdot \operatorname{sign}(x-b)|x-b|^{\frac{1}{k}}$. Let $\ell, u \in \mathbb{R}$ with $\ell < u$. Let $g : \mathbb{R} \to \mathbb{R}$ be a function defined by $g(x) := f(u+x) - f(\ell+x)$. Then g has a global maximum in $x_0 = b - \frac{\ell+u}{2}$.*

Proof: If $k = 1$, then $g(x) = c \cdot \operatorname{sign}(u+x-b)|u+x-b| - c \cdot \operatorname{sign}(\ell+x-b)|\ell+x-b| = c(u-\ell)$, hence g is a constant function.

If $k > 1$, then the first derivate of g is given by $g'(x) = \frac{c}{k}(|u+x-b|^{\frac{1}{k}-1} - |\ell+x-b|^{\frac{1}{k}-1})$. Hence its unique critical point is in $x_0 = b - \frac{\ell+u}{2}$. Its value is $g(b - \frac{\ell+u}{2}) = 2c\cdot\operatorname{sign}(u-l)|\frac{u-l}{2}|^{\frac{1}{k}}$. To see that it is a maximum we check two points, one left and one right of x_0: $g(b-\ell) = f(u+b-\ell) - f(b) = c\cdot\operatorname{sign}(u-\ell)|u-\ell|^{\frac{1}{k}} < g(x_0)$ and $g(b-u) = f(b) - f(\ell+b-u) = -c\cdot\operatorname{sign}(\ell-u)|\ell-u|^{\frac{1}{k}} = c\cdot\operatorname{sign}(u-\ell)|u-\ell|^{\frac{1}{k}} < g(x_0)$. □

Lemma 3. *For a node v_i the error $\varepsilon_{j,i}(v_i)$ in iteration $\langle j, i\rangle$ is bounded from above by a function in $\pi'_{v_i} - \pi_{v_i}$. In detail, we have that*

$$\varepsilon_{j,i}(v_i) \leq \left(\sum_{e\in E'} 2\alpha_e^{-\frac{1}{k_e+1}}\right) \cdot |\pi'_{v_i} - \pi_{v_i}|^{\frac{1}{k+1}}. \tag{5.6}$$

Proof: Let $\mathcal{E}_{j,i}$ denote the total error $\mathcal{E}$ in iteration $\langle j, i\rangle$. Let π_{v_i} be the potential value before the update process at iteration $\langle j, i\rangle$ and π'_{v_i} the potential value at iteration $\langle j, i+1\rangle$, that is, after the update in iteration $\langle j, i\rangle$ has been processed. For $e \in \delta(v_i)$ consider

$$q'_e - q_e = \alpha_e^{-\frac{1}{k_e+1}} \operatorname{sign}(\pi'_{v_i} - \pi_w - \tilde{\beta}_e)|\pi'_{v_i} - \pi_w - \tilde{\beta}_e|^{\frac{1}{k_e+1}} - \alpha_e^{-\frac{1}{k_e+1}} \operatorname{sign}(\pi_{v_i} - \pi_w - \tilde{\beta}_e)|\pi_{v_i} - \pi_w - \tilde{\beta}_e|^{\frac{1}{k_e+1}}. \tag{5.7}$$

We apply Lemma 2. For this, let $c := \alpha_e^{-\frac{1}{k_e+1}}, b := \tilde{\beta}_e, k := k_e + 1, u := \pi'_{v_i} - \pi_w$ and $\ell := \pi_{v_i} - \pi_w$. Hence the right-hand side of (5.7) equals the function g in Lemma 2 for $x = 0$:

$$q'_e - q_e = g(0). \tag{5.8}$$

Let $x_0 := \tilde{\beta}_e - \frac{(\pi_{v_i}-\pi_w)+(\pi'_{v_i}-\pi_w)}{2} = \tilde{\beta}_e - \frac{\pi_{v_i}+\pi'_{v_i}-2\pi_w}{2}$. Using first the result of this lemma, and then $2^{\frac{k_e}{k_e+1}} \leq 2$, we obtain the estimation

$$q'_e - q_e \leq g(x_0) \leq 2\alpha_e^{-\frac{1}{k_e+1}} \operatorname{sign}(\pi'_{v_i} - \pi_{v_i})|\pi'_{v_i} - \pi_{v_i}|^{\frac{1}{k_e+1}}. \tag{5.9}$$

Note that the algorithm only considers those nodes v_i with a positive error, $\varepsilon(v_i) > 0$. In order to reduce this error, the node potential has to be increased, that is, $\pi'_{v_i} > \pi_{v_i}$. Hence $\operatorname{sign}(\pi'_{v_i} - \pi_{v_i}) > 0$. From this it follows that

$$\begin{aligned}
\varepsilon_{j,i}(v_i) &= \sum_{e\in\delta(v_i)} (q'_e - q_e) && \text{(5.10a)}\\
&\leq \sum_{e\in\delta(v_i)} 2\alpha_e^{-\frac{1}{k_e+1}}|\pi'_{v_i} - \pi_{v_i}|^{\frac{1}{k_e+1}} && \text{(5.10b)}\\
&\leq \left(\sum_{e\in E'} 2\alpha_e^{-\frac{1}{k_e+1}}\right) \cdot |\pi'_{v_i} - \pi_{v_i}|^{\frac{1}{k+1}}. && \text{(5.10c)}
\end{aligned}$$

Here we set k such that $|\pi'_{v_i} - \pi_{v_i}|^{\frac{1}{k+1}} = \min\{|\pi'_{v_i} - \pi_{v_i}|^{\frac{1}{k_e+1}} : e \in E\}$ □

Lemma 4. *There exists an upper bound $M \in \mathbb{R}_+$ on the maximum potential difference $|\pi_v - \pi_w|$ that occurs on any arc $e = (v, w) \in E'$ during the execution of the algorithm.*

Proof: Let $e = (v, w) \in E'$. Let M_e be an upper bound for the potential value difference between π_v and π_w during the execution of the algorithm, that is, $M_e := \sup\{|\pi_v^{\langle j,i\rangle} - \pi_w^{\langle j,i\rangle}| : j = 0, 1, 2, \dots\ i = 1, 2, \dots, |V|\}$, where $\pi_v^{\langle j,i\rangle}, \pi_w^{\langle j,i\rangle}$ is the value of π_v, π_w in step $\langle j, i\rangle$, respectively. We show that M_e is finite.

First we assume $\tilde{\beta}_e = 0$ for all $e \in E'$. Then an upper bound M_e on the maximum node potential difference can be derived by considering an artificial arc $e = (v, w)$ with $\alpha_e \in \arg\max\{\alpha_e : e \in E'\}$ and $k_e \in \arg\max\{k_e : e \in E'\}$, where the entire network flow $\sum_{v\in V} |s_v/2| =: q_e$ is flowing over this particular arc. Then $\pi_v - \pi_w = \alpha_e q_e |q_e|^{k_e} =: M_e$ is the desired upper bound.

In the general case (when $\tilde{\beta}_e \neq 0$ for some or all $e \in E'$). Then an upper bound on the maximum node potential difference can be derived by considering an artificial arc $e = (v, w)$ as before, and its artificial anti-parallel counterpart arc $e' = (w, v)$, with $\hat{\beta}_e := \hat{\beta}_{e'} := \sum_{e\in E'} |\tilde{\beta}_e|$ and $\alpha_{e'} := 0$. Consider the equation system for e and e', i.e., $\alpha_e q_e |q_e| - \hat{\beta}_e = \hat{\beta}_{e'}$, where $q_e := \sum_{v\in V} |s_v/2| + q_{e'}$. From this equation we can determine $q_{e'}$, and thus q_e, which again gives an upper bound $\pi_v - \pi_w = \alpha_e q_e |q_e|^{k_e} =: M_e$.

Taking the maximum over all M_e we define $M := \max\{M_e \mid e \in E'\}$. □

Theorem 2. *Let $n := |V|$. Let $\mathcal{E}_j$ denote the error $\mathcal{E}$ after j iterations of the outer loop 1. Then the total error in Algorithm 1 is strictly monotone decreasing, and there exist constants $k, C \geq 0$ such that*

$$\mathcal{E}_{j+1} \leq \mathcal{E}_j - C^n \left(\frac{\mathcal{E}_j}{2n}\right)^k.$$

Proof: We consider an arbitrary outer iteration j of the algorithm. Then the algorithm selects a node v_1 with maximal positive error.

We show that $\varepsilon_{j,1}(v_1) \geq \mathcal{E}_{j,1}/(2n) = \mathcal{E}_j/(2n)$. Assume that $\varepsilon_{j,1}(v_1) < \mathcal{E}_{j,1}/(2n)$. Since v_1 was chosen as arg max, we have that $\varepsilon_{j,1}(v) < \mathcal{E}_{j,1}/(2n)$ for all other nodes $v \in V$ with $\varepsilon_{j,1}(v) > 0$. Then we sum over all these nodes and obtain

$$\sum_{\substack{v\in V,\\ \varepsilon_{j,1}(v)>0}} \varepsilon_{j,1}(v) < \sum_{\substack{v\in V,\\ \varepsilon_{j,1}(v)>0}} \frac{\mathcal{E}_{j,1}}{2n} \leq \sum_{v\in V} \frac{\mathcal{E}_{j,1}}{2n} = \frac{\mathcal{E}_{j,1}}{2}. \tag{5.11}$$

Note that $\mathcal{E}_{j,1} = \sum_{v\in V} |\varepsilon_{j,1}(v)|$ and $\sum_{v\in V} \varepsilon_{j,1}(v) = 0$, hence $\mathcal{E}_{j,1}/2 = \sum_{v\in V, \varepsilon_{j,1}(v)>0} \varepsilon_{j,1}(v)$, which is a contradiction to the estimation in (5.11).

Consider an arc $e = (v_1, w)$. Let $f(x)$ be defined as in Lemma 2 with $c := \alpha_e^{-\frac{1}{k_e+1}}, b := \tilde{\beta}_e, k := k_e + 1$. Then $f'(x) = \frac{c}{k}|x-b|^{1/k-1}$ for $x \in \mathbb{R}\backslash\{b\}$. f' is a monotone decreasing on $]b, \infty[$ and monotone increasing on $]-\infty, b[$. Denote by q_e, π_{v_1} the flow and potential value in step $\langle j, 0\rangle$, and by q'_e, π'_{v_1} the flow and potential value in step $\langle j, 1\rangle$ (i.e., after the update). Since $\varepsilon_{j,0}(v_1) > 0$ we increase the potential: $\pi_{v_1} < \pi'_{v_1}$ and consequently $q_e < q'_e$. The flow difference $q'_e - q_e$ can be expressed by the mean value theorem: There exists a $\xi \in]\pi_{v_1} - \pi_w, \pi'_{v_1} - \pi_w[$ such that $q'_e - q_e = f(\pi'_{v_1} - \pi_w) - f(\pi_{v_1} - \pi_w) = f'(\xi)(\pi'_{v_1} - \pi_{v_1})$. Let M be defined as in the proof of Lemma 4. Since $|\xi| \le M$ we can estimate $q'_e - q_e$ from below as

$$q'_e - q_e \ge f'(M)(\pi'_{v_1} - \pi_{v_1}). \tag{5.12}$$

From the estimation in Lemma 3 we deduce that

$$\left(\sum_{\tilde{e}\in E'} 2\alpha_{\tilde{e}}^{-\frac{1}{k_{\tilde{e}}+1}}\right)^{-k-1} \varepsilon_{j,1}(v_1)^{k+1} \le \pi'_{v_1} - \pi_{v_1}. \tag{5.13}$$

Putting (5.12) and (5.13) together, we get

$$f'(M)\left(\sum_{\tilde{e}\in E'} 2\alpha_{\tilde{e}}^{-\frac{1}{k_{\tilde{e}}+1}}\right)^{-k-1} \varepsilon_{j,1}(v_1)^{k+1} \le q'_e - q_e. \tag{5.14}$$

Since f' is strictly monotone decreasing on $]b, \infty[$, we can chose $M' \ge M$ such that

$$C_e := f'(M')\left(\sum_{\tilde{e}\in E'} 2\alpha_{\tilde{e}}^{-\frac{1}{k_{\tilde{e}}+1}}\right)^{-k-1} \le 1. \tag{5.15}$$

Note that we can repeat the arguments above for any arc $e \in E'$ in the network. In order to define a constant independent of arc e, we set

$$C := \min\{C_e \mid e \in E'\} \le 1. \tag{5.16}$$

The inner loop 2 of the algorithm orders the nodes in such way that the potential for those nodes with a positive error is only increased once during the execution of this loop. In each inner iteration a node is selected which is a neighbor of already updated nodes. If this node v_i has a positive error, then the same reasoning carried out for v_1 hold also for this node. We can estimate the amount of flow that is pushed through the network, until a node with a negative error is reached. Here the additional flow (coming from a node with a positive error) and the negative error partially cancels out. By Lemma 1 the total error cannot increase in the following inner iterations. Hence in the end of the inner loop, the total error strictly decreases. The amount of decrease is estimated in the following.

In equation (5.14) the amount of flow that was pushed from node v_1 to some neighbor v_m was estimated to be at least $C \cdot \varepsilon_{j,1}(v_1)^{k_e+1}$. If $\varepsilon_{j,1}(v_1) \le 1$, let $k := \max\{k_e : e \in E'\} + 1$, otherwise $k := \min\{k_e : e \in E'\} + 1$. Then the amount of flow from node v_1 to v_m is at least $C \cdot \varepsilon_{j,1}(v_1)^k$. Assume that v_m also has a positive error. Then the algorithm can select v_m as v_2, and by repeating the same arguments from above, we can estimate that the amount of flow that is pushed from v_2 to one of its neighbors $v_{m'}$ (not v_1) is at least $C \cdot \left(C \cdot \varepsilon_{j,1}(v_1)^k\right) = C^2 \cdot \varepsilon_{j,1}(v_1)^k$. In the worst case, every node has a positive error, so the initial error in v_1 is pushed through the whole network. From one node to the neighbor, the initial amount of flow decreases to $C^n \cdot \varepsilon_{j,1}(v_1)^k$, where $n := |V|$. The very last node, however, has a negative error of at most $-\varepsilon_{j,1}(v_1)$ (or less). Hence during this whole inner iteration, an error of at least $C^n \cdot \varepsilon_{j,1}(v_1)^k$ cancels. Note that $C^n \cdot \varepsilon_{j,1}(v_1)^k \ge C^n \cdot (\mathcal{E}_j/(2n))^k$. Hence the total error is guaranteed to decrease by at least $C^n \cdot (\mathcal{E}_j/(2n))^k$ in one outer iteration. □

5.5 Solving KKT Systems

We use the definition of Boyd and Vandenberghe [4] to characterize the relaxed leaf problem (5.1). They distinguish *convex optimization problems* from *abstract convex optimization problems*. The first type of problem is defined by a convex objective function and convex constraints, whereas the second type also uses a convex objective, but a convex feasible region. So the constraints are not required to be convex in an abstract convex optimization problem. Clearly, a convex optimization problem is also an abstract convex optimization problem. In this sense, problem (5.1) is an abstract convex problem: The convexity of the feasible solution space for (q, π) follows from Collins et al. [5] (c.f. Section 5.1) and Theorem 1, which states that the solution space is a non-empty affine subspace. The flow is unique, while the feasible vectors π form a straight line. The additional constraints (5.1c)-(5.1f) are linear and do not disturb the feasibility (i.e., rendering the problem infeasible). So in total, problem (5.1) is an abstract convex optimization problem over a non-empty set of feasible solutions. Hence we can use an NLP solver that guarantees only local optimality (such as IPOPT, for instance) to solve the relaxed leaf problem (5.1) already to global optimality.

Interpretation of Lagrange dual variables In order to understand the infeasibility of a leaf problem we are going to analyze the Lagrange dual parameters of the Lagrange dual of the relaxed leaf problem (5.1) at a KKT point. Similar to a dual vector in linear programming, this yields a certificate for infeasibility. Denote by $(\mu, \lambda) = (\mu_v, \mu_e, \lambda_v^+, \lambda_v^-, \lambda_e^+, \lambda_e^-, \lambda_v, \lambda_e)$, such that $\mu_v, \mu_e \in \mathbb{R}$ and $\lambda_v^+, \lambda_v^-, \lambda_v, \lambda_e, \lambda_e^+, \lambda_e^- \in \mathbb{R}_+$ the Lagrange multipliers. Then the Lagrangian of problem (5.1) has the form

$$\begin{aligned} L(q, \pi, \Delta, \mu, \lambda) = & \sum_{v \in V} \Delta_v + \sum_{e \in E'} \Delta_e \\ & + \sum_{e \in E'} \mu_e \left(\Phi_e(q_e) - (\pi_v - \gamma_e \pi_w) \right) \\ & + \sum_{v \in V} \mu_v \left(s_v - \sum_{e \in \delta^+(v)} q_e + \sum_{e \in \delta^-(v)} q_e \right) \\ & + \sum_{v \in V} \left(\lambda_v^+ (\pi_v - \Delta_v - \overline{\pi}_v) + \lambda_v^- (\underline{\pi}_v - \pi_v - \Delta_v) \right) \\ & + \sum_{e \in E'} \left(\lambda_e^+ (q_e - \Delta_e - \overline{q}_e) + \lambda_e^- (\underline{q}_e - q_e - \Delta_e) \right) \\ & - \sum_{v \in V} \lambda_v \Delta_v - \sum_{e \in E'} \lambda_e \Delta_e. \end{aligned} \tag{5.17}$$

For a local optimum of a nonlinear problem it is shown by Boyd and Vandenberghe [4] that there exist values for these dual variables fulfilling the KKT conditions. From these KKT conditions we derive the following constraints, c.f. equation system (2.1):

$$\frac{\partial L}{\partial q_e}: \quad \mu_e \left(\nabla_{q_e} \Phi_e(q_e) \right) + \lambda_e^+ - \lambda_e^- = \mu_v - \mu_w \qquad \forall\, e \in E', \tag{5.18a}$$

$$\frac{\partial L}{\partial \pi_v}: \quad \sum_{e \in \delta^+(v)} \mu_e - \sum_{e \in \delta^-(v)} \mu_e \gamma_e = \lambda_v^+ - \lambda_v^- \qquad \forall\, v \in V, \tag{5.18b}$$

$$\frac{\partial L}{\partial \Delta_v}: \quad \lambda_v^+ + \lambda_v^- + \lambda_v = 1 \qquad \forall\, v \in V, \tag{5.18c}$$

$$\frac{\partial L}{\partial \Delta_e}: \quad \lambda_e^+ + \lambda_e^- + \lambda_e = 1 \qquad \forall\, e \in E'. \tag{5.18d}$$

From this we conclude:

$$\pi_v < \overline{\pi}_v \Rightarrow \lambda_v^+ = 0, \quad \pi_v = \overline{\pi}_v \Rightarrow 0 \le \lambda_v^+ \le 1, \quad \pi_v > \overline{\pi}_v \Rightarrow \lambda_v^+ = 1 \qquad \forall\, v \in V, \tag{5.19a}$$

$$\pi_v > \underline{\pi}_v \Rightarrow \lambda_v^- = 0, \quad \pi_v = \underline{\pi}_v \Rightarrow 0 \le \lambda_v^- \le 1, \quad \pi_v < \underline{\pi}_v \Rightarrow \lambda_v^- = 1 \qquad \forall\, v \in V, \tag{5.19b}$$

$$q_e < \overline{q}_e \Rightarrow \lambda_e^+ = 0, \quad q_e = \overline{q}_e \Rightarrow 0 \le \lambda_e^+ \le 1, \quad q_e > \overline{q}_e \Rightarrow \lambda_e^+ = 1 \qquad \forall\, e \in E', \tag{5.19c}$$

$$q_e > \underline{q}_e \Rightarrow \lambda_e^- = 0, \quad q_e = \underline{q}_e \Rightarrow 0 \le \lambda_e^- \le 1, \quad q_e < \underline{q}_e \Rightarrow \lambda_e^- = 1 \qquad \forall\, e \in E'. \tag{5.19d}$$

The interpretation of this constraint system is as follows. The second equality (5.18b) indicates, that μ_e represent a network flow in G' where each edge $e \in E'$ has μ_e as its flow variable. The in- and out-flows at sources and sinks are given by $\lambda_v^+ - \lambda_v^-$, and the relation of these values with the arc flows is given by the weighted flow conservation (5.18b) (also called generalized flow conservation, see [23] and the references therein). Thus the node flow must not necessarily be balanced, i.e., $\sum_{v \in V}(\lambda_v^+ - \lambda_v^-) \neq 0$. The implications (5.19a) and (5.19b) ensure that a non-zero entry flow is only allowed, if $\pi_v \geq \overline{\pi}_v$. Furthermore, a non-zero exit flow can only occur at a node fulfilling $\pi_v \leq \underline{\pi}_v$. Looking at equation (5.18a), the dual value μ_v can be interpreted as a dual potential at node v. The values λ_e^+, λ_e^- enforce a dual decrease or increase of the potential values and so react like a dual active element (compressor or control valve), c.f. (5.19c) and (5.19d).

The KKT system (5.18) plays a role similar to the Max-Flow-Min-Cut theorem in classical linear network flow theorem of Ford and Fulkerson [14]. Let us recall that a network flow without node potentials is bounded by any cut in the network, and that the flow value of a maximum flow equals the capacity of a minimum cut. Consider a flow that is locally infeasible, i.e., violates upper capacity bounds on some arc $e = (v, w)$. Then we can try to make it feasible by identifying a cycle in the graph that contains this arc e, and has some residual capacities. We can reduce the flow on arc e and increase it on the remaining part of the cycle. Such cycle exists, if and only if the flow value is less-or-equal that minimum cut value (by the Ford-Fulkerson theorem). Now coming back to our network flow problem with node potentials. Assume we have a given flow and corresponding node potential values which violate the upper bounds at some node v. (W.l.o.g. there a no lower bounds violated, because we can raise all potential values simultaneously according to Theorem 1.) Then there exists a path P from v to some node w with $\sum_{e \in P} \Phi_e(q_e) > \overline{\pi}_v - \underline{\pi}_w$. We can also try to make it feasible by identifying a path P' in the graph from v to some other node w with $\sum_{e \in P'} \Phi_e(q_e) < \overline{\pi}_v - \underline{\pi}_w$. This path also has a remaining capacity (similar to the cycle in the Ford-Fulkerson case), so that we increase the flow on P' and decrease it on P. After this decrease on P the node potential value in v can also be reduced. Such path exists, if and only if there is no path from v to w in the dual flow defined by system (5.18). The relation of primal and dual flow values for the case of remaining node potentials is shown in Figure 5.2 on the left, and the case of violated node potential bounds is shown on the right.

Figure 5.3 shows a visualization of a dual flow in a test network of practical dimension (`net1a`). There are four dual entries (marked as large diamonds), and four dual exits marked as large squares. All other internal nodes are marked as small diamonds. The color of a node corresponds to the dual node potential. The arc width represents the dual flow value (the thicker the more dual flow), while its colors depicts the difference of the dual node potential at both end nodes. The figure shows an infeasible (primal) flow, where the entries node potentials are above their respective upper limit, and the exit node potentials are below their respective lower limit. (Note that there are more primal entries and exits; only the four exceeding the bounds became dual entries and exits.) One can try to shift the potentials towards feasibility by considering a dual path between a single dual entry and a single dual exit. Trying to reduce the potential excess on this path means to increase it in other parts of the network, so that one can find a new path with the same potential excess, which is again a certificate for infeasibility. Hence no feasible primal solution exists.

6 Relaxation of Flow Conservation Constraints

In this section we consider a modification of a leaf problem. Basically it is a relaxation of the flow conservation constraint by slack variables.

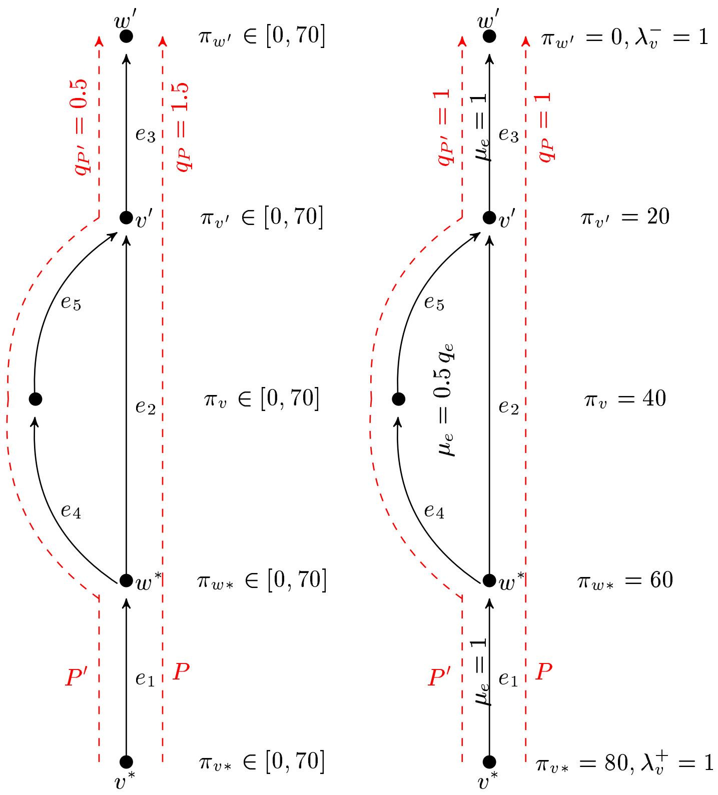
$$\min \sum_{v \in V} \left(\Delta_v^+ + \Delta_v^-\right) + \sum_{e \in E'} \left(\Delta_e^+ + \Delta_e^-\right) \quad \text{s. t.} \tag{6.1a}$$

$$\alpha_e q_e |q_e|^{k_e} + \tilde{\beta}_e - (\pi_v - \gamma_e \pi_w) = 0 \qquad \forall\, e = (v, w) \in E', \tag{6.1b}$$

$$\sum_{e \in \delta^+(v)} (q_e - (\Delta_e^+ - \Delta_e^-)) - \sum_{e \in \delta^-(v)} (q_e - (\Delta_e^+ - \Delta_e^-)) - (\Delta_v^+ - \Delta_v^-) = s_v \qquad \forall\, v \in V, \tag{6.1c}$$

$$\pi_v \leq \overline{\pi}_v \qquad \forall\, v \in V, \tag{6.1d}$$

$$\pi_v \geq \underline{\pi}_v \qquad \forall\, v \in V, \tag{6.1e}$$



$\alpha_{e_1} = 5, \alpha_{e_2} = 40, \alpha_{e_3} = 5, \alpha_{e_4} = 20, \alpha_{e_5} = 20$

Left: $\overline{\pi}_{v^*} - \underline{\pi}_{w'} > \sum_{e \in P'} \Phi_e(q_e) < \sum_{e \in P} \Phi_e(q_e) > \overline{\pi}_{v^*} - \underline{\pi}_{w'}$

Right: $\overline{\pi}_{v^*} - \underline{\pi}_{w'} < \sum_{e \in P'} \Phi_e(q_e) = \sum_{e \in P} \Phi_e(q_e)$

Fig. 5.2. Relation of primal and dual flows. The left picture shows arc flows which are not optimal. In the right picture the arc flows form a KKT solution. The dashed lines indicate the flow along the marked paths.

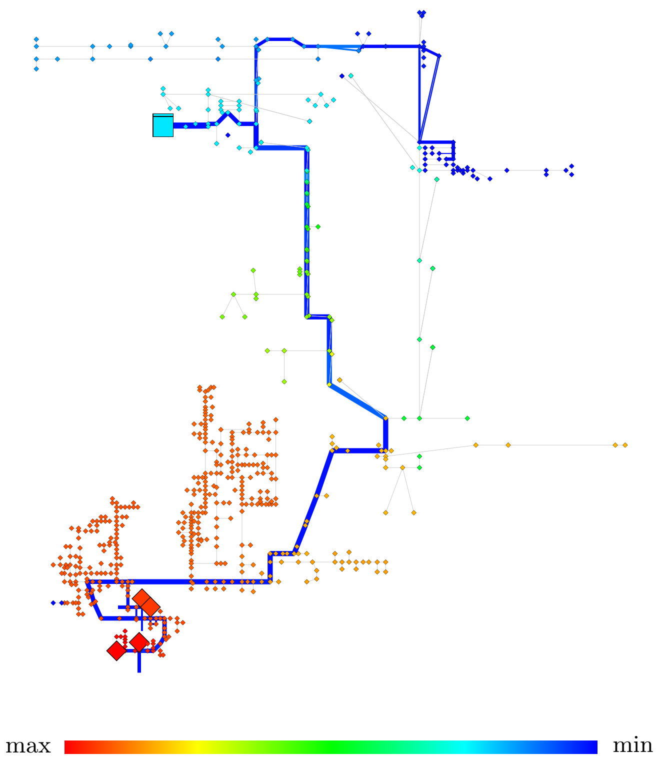


Fig. 5.3. The Lagrange dual variables of a KKT point for test network `net1a`.

$$q_e \leq \overline{q}_e \quad \forall\, e \in E', \tag{6.1f}$$

$$q_e \geq \underline{q}_e \quad \forall\, e \in E', \tag{6.1g}$$

$$\Delta_v^- \, (\overline{\pi}_v - \pi_v) = 0 \quad \forall\, v \in V, \tag{6.1h}$$

$$\Delta_v^+ \, (\pi_v - \underline{\pi}_v) = 0 \quad \forall\, v \in V, \tag{6.1i}$$

$$\Delta_e^- \, (\overline{q}_e - q_e) = 0 \quad \forall\, e \in E', \tag{6.1j}$$

$$\Delta_e^+ \, (q_e - \underline{q}_e) = 0 \quad \forall\, e \in E', \tag{6.1k}$$

$$\pi_v \in \mathbb{R} \quad \forall\, v \in V, \tag{6.1l}$$

$$q_e \in \mathbb{R} \quad \forall\, e \in E', \tag{6.1m}$$

$$\Delta_v^\pm \in \mathbb{R}_+ \quad \forall\, v \in V, \tag{6.1n}$$

$$\Delta_e^\pm \in \mathbb{R}_+ \quad \forall\, e \in E'. \tag{6.1o}$$

On an arc $e \in E'$ a positive slack value $\Delta_e^+ > 0$ or $\Delta_e^- > 0$ is feasible only if the flow variable q_e reaches its bounds $\underline{q}_e$ or $\overline{q}_e$, respectively. Accordingly, a positive slack value $\Delta_v^+ > 0$ or $\Delta_v^- > 0$ at a node $v \in V$ is feasible only if the potential value π_v attains a boundary value $\underline{\pi}_v$ or $\overline{\pi}_v$, respectively.

Problem (6.1) can be infeasible. This happens, if the flow bounds (6.1f) and (6.1g) enforce such a high amount of flow on an arc, that the potential loss (as deduced by equation (6.1b) is in conflict with the pressure bounds on both end nodes of the arc. This situation can easily be detected in a preprocessing step by checking one arc after the other separately. If the problem turns out to be infeasible, we stop, because the leaf problem is infeasible. Otherwise, we can proceed with its solution.

6.1 Existence of a solution

To compute a solution to (6.1) we extend the convex optimization problem (5.2) proposed by Collins et al. [5]. We introduce slack variables and add further terms to the objective function. Then this extension is of the following form:

$$\min_{q,\Delta} \left(\sum_{e\in E'} \int_{q_e^0}^{q_e} \varPhi_e(t)\,\mathrm{d}t + \sum_{v\in V} \left(\overline{\pi}_v\, \Delta_v^- - \underline{\pi}_v\, \Delta_v^+\right) + \sum_{e\in E'} \left(\varPhi_e(\overline{q}_e)\,\Delta_e^- - \varPhi_e(\underline{q}_e)\,\Delta_e^+\right)\right) \text{ s.t.} \tag{6.2a}$$

$$s_v - \sum_{e\in\delta^+(v)} \left(q_e - (\Delta_e^+ - \Delta_e^-)\right) + \sum_{e\in\delta^-(v)} \left(q_e - (\Delta_e^+ - \Delta_e^-)\right) + (\Delta_v^+ - \Delta_v^-) = 0\,\forall\, v\in V, \tag{6.2b}$$

$$\Delta_v^\pm \geq 0\,\forall\, v\in V, \tag{6.2c}$$

$$\Delta_e^\pm \geq 0\,\forall\, e\in E'. \tag{6.2d}$$

As before, q_e^0 is the root of $\varPhi_e(\cdot)$. In the next theorem we characterize this nonlinear optimization problem by analyzing the KKT conditions of this constraint system.

Theorem 3. *The nonlinear program* (6.2) *is convex. Every optimal solution to* (6.2) *can be transformed to a feasible solution for* (6.1)*, if* $\gamma_e = 1$ *for all arcs* $e\in E'$*.*

Later, in Theorem 5, we will show that this feasible solution is in fact an optimal one.
Proof: Problem (6.2) is a convex problem, as $\varPhi_e$ is a monotone increasing function and the constraints are of linear type. To analyse the optimal points, we analyze its KKT points. Its Lagrange formulation with the dual variables $\mu_v \in \mathbb{R}$ and $\lambda_v^\pm, \lambda_e^\pm \in \mathbb{R}_+$ for $v\in V$ and $e\in E'$ is as follows:

$$\begin{aligned}
L(q,\Delta,\mu,\lambda) = {} & \sum_{e\in E'} \int_{q_e^0}^{q_e} \varPhi_e(t)\,\mathrm{d}t + \sum_{v\in V} \left(\overline{\pi}_v\, \Delta_v^- - \underline{\pi}_v\, \Delta_v^+\right) + \sum_{e\in E'} \left(\varPhi_e(\overline{q}_e)\,\Delta_e^- - \varPhi_e(\underline{q}_e)\,\Delta_e^+\right) \\
& + \sum_{v\in V} \mu_v \left(s_v - \sum_{e\in\delta^+(v)} \left(q_e - (\Delta_e^+ - \Delta_e^-)\right) + \sum_{e\in\delta^-(v)} \left(q_e - (\Delta_e^+ - \Delta_e^-)\right) + (\Delta_v^+ - \Delta_v^-)\right) \\
& - \sum_{v\in V} \left(\lambda_v^+ \Delta_v^+ + \lambda_v^- \Delta_v^-\right) \\
& - \sum_{e\in E'} \left(\lambda_e^+ \Delta_e^+ + \lambda_e^- \Delta_e^-\right).
\end{aligned}$$

We define $\pi_v := \mu_v$ for $v\in V$. The KKT condition $\nabla_{(q,\Delta)} L = 0$ results in the following constraints:

$$\frac{\partial L}{\partial q_e}: \quad \varPhi_e(q_e) = \pi_v - \pi_w \quad \forall\, e=(v,w)\in E', \tag{6.3a}$$

$$\frac{\partial L}{\partial \Delta_v^-}: \quad \overline{\pi}_v - \lambda_v^- = \pi_v \quad \forall\, v\in V, \tag{6.3b}$$

$$\frac{\partial L}{\partial \Delta_v^+}: \quad \underline{\pi}_v + \lambda_v^+ = \pi_v \quad \forall\, v\in V, \tag{6.3c}$$

$$\frac{\partial L}{\partial \Delta_e^-}: \quad \varPhi_e(\overline{q}_e) - \lambda_e^- = \pi_v - \pi_w \quad \forall\, e=(v,w)\in E', \tag{6.3d}$$

$$\frac{\partial L}{\partial \Delta_e^+}: \quad \varPhi_e(\underline{q}_e) + \lambda_e^+ = \pi_v - \pi_w \quad \forall\, e=(v,w)\in E'. \tag{6.3e}$$

From (6.3a) it follows directly that the solution (q,π,Δ) satisfies constraint (6.1b). Using (6.3a)–(6.3e) together with the complementarity slackness conditions

$$\lambda_v^+\,\Delta_v^+ = 0 \quad \text{and} \quad \lambda_v^-\,\Delta_v^- = 0,$$

and

$$\lambda_e^+ \Delta_e^+ = 0 \quad \text{and} \quad \lambda_e^- \Delta_e^- = 0,$$

we observe that the following constraints are fulfilled for a primal solution:

$$\Delta_v^-(\overline{\pi}_v - \pi_v) = 0,$$
$$\Delta_v^+(\pi_v - \underline{\pi}_v) = 0,$$

and, using the strict monotonicity of Φ_e,

$$\Delta_e^-(\Phi_e(\overline{q}_e) - \Phi_e(q_e)) = 0 \quad \Rightarrow \quad \Delta_e^-(\overline{q}_e - q_e) = 0,$$
$$\Delta_e^+(\Phi_e(q_e) - \Phi_e(\underline{q}_e)) = 0 \quad \Rightarrow \quad \Delta_e^+(q_e - \underline{q}_e) = 0.$$

Hence constraints (6.1h)–(6.1k) are fulfilled. Remember that $\lambda \geq 0$. Therefore, (6.1d)–(6.1g) are also fulfilled. Note that the flow conservation constraints (6.1c) are automatically fulfilled due to constraints (6.2a). Altogether, (q, π, Δ) is a feasible solution for (6.1). □

In order to give an alternative procedure for computing a feasible solution to (6.1), we consider an extension of (5.3). We consider an arbitrary node r of our graph as a root node and denote a path from r to node $v \in V$ by $P_r(v)$. We set $\gamma_{r,v}$ defined as $\gamma_{r,v} := \prod_{e \in P_r(v)} \gamma_e$. Then the extension is as follows:

$$\min_{\pi} \sum_{e=(v,w)\in E'} \gamma_{r,v} \int_{\Delta_e^0}^{\pi_v - \gamma_e \pi_w} \Phi_e^{-1}(t)\mathrm{d}t - \sum_{v\in V} \gamma_{r,v} \int_0^{\pi_v} s_v \mathrm{d}t \quad \text{s.t.} \tag{6.4a}$$

$$\gamma_{r,v}\underline{\pi}_v \leq \gamma_{r,v}\pi_v \leq \gamma_{r,v}\overline{\pi}_v \qquad \forall\, v \in V, \tag{6.4b}$$

$$\gamma_{r,v}\Phi_e(\underline{q}_e) \leq \gamma_{r,v}(\pi_v - \gamma_e \pi_w) \leq \gamma_{r,v}\Phi_e(\overline{q}_e) \qquad \forall\, e = (v,w) \in E'. \tag{6.4c}$$

Here Δ_e^0 is the root of the function $\Phi_e^{-1}(\cdot)$, the inverse of $\Phi_e(\cdot)$. We note that the value $\gamma_{r,v} = \prod_{e \in P_r(v)} \gamma_e$ does not depend on the choice of the path connecting the nodes r and v.

Theorem 4. *The nonlinear optimization problem* (6.4) *is convex. Its optimum solution yields a feasible solution for* (6.1)*.*

Proof: We note that the constraints of (6.4) are of linear type. The objective function is convex, because of the definition of Δ_e^0. Hence (6.4) is convex.

Now we analyze the solution of (6.4). The Lagrange formulation with the dual variables $\Delta_v^\pm \in \mathbb{R}_+$ for each node $v \in V$ and $\Delta_e^\pm \in \mathbb{R}_+$ for each arc $e \in E'$ is as follows:

$$\begin{aligned} L(\pi, \Delta) = &\sum_{e=(v,w)\in E'} \gamma_{r,v} \int_{\Delta_e^0}^{\pi_v - \gamma_e \pi_w} \Phi_e^{-1}(t)\mathrm{d}t - \sum_{v\in V} \gamma_{r,v} \int_0^{\pi_v} s_v \mathrm{d}t \\ &+ \sum_{v\in V} \Delta_v^- \gamma_{r,v}(\pi_v - \overline{\pi}_v) + \Delta_v^+ \gamma_{r,v}(\underline{\pi}_v - \pi_v) \\ &+ \sum_{e=(v,w)\in E'} \Delta_e^- \gamma_{r,v}(\pi_v - \gamma_e \pi_w - \Phi_e(\overline{q}_e)) \\ &+ \sum_{e=(v,w)\in E'} \Delta_e^+ \gamma_{r,v}(\Phi_e(\underline{q}_e) - (\pi_v - \gamma_e \pi_w)). \end{aligned}$$

For an optimal solution π we define q by

$$q_e := \Phi_e^{-1}(\pi_v - \gamma_e \pi_w) \quad \Leftrightarrow \quad \Phi_e(q_e) = \pi_v - \gamma_e \pi_w \quad \forall\, e = (v,w) \in E'. \tag{6.5}$$

We consider the vector (q, π, Δ), and show that it is a solution of (6.1). By the definition of q in (6.5) constraints (6.1b) are fulfilled. The Karush-Kuhn-Tucker Theorem ensures that

$$\frac{\partial L}{\partial \pi_v}: \quad \sum_{e\in\delta^+(v)} \gamma_{r,v}(q_e - (\Delta_e^+ - \Delta_e^-)) - \sum_{e\in\delta^-(v)} \gamma_e \gamma_{r,w}(q_e - (\Delta_e^+ - \Delta_e^-)) - \gamma_{r,v}(\Delta_v^+ - \Delta_v^-) = \gamma_{r,v} s_v \qquad \forall\, v \in V.$$

As $\gamma_e \gamma_{r,w} = \gamma_{r,v}$ for all arcs $e = (w,v) \in E'$, we obtain that the flow conservation constraints (6.1c) are satisfied. Constraints (6.1d) and (6.1e) are fulfilled, because of constraints (6.4b). Constraints (6.1f) and (6.1g) are fulfilled, because of constraints (6.4c) and the strictly monotonicity (hence bijectivity) of Φ_e and Φ_e^{-1} (remember $\gamma_{r,v} \neq 0$). We observe from the complementary slackness conditions that the following constraints are fulfilled:

$$\begin{aligned} \Delta_v^-(\overline{\pi}_v - \pi_v) &= 0, \qquad \Delta_v^+(\pi_v - \underline{\pi}_v) = 0 \qquad \forall\, v \in V, \\ \Delta_e^-(\overline{q}_e - q_e) &= 0, \qquad \Delta_e^+(q_e - \underline{q}_e) = 0 \qquad \forall\, e \in E', \end{aligned}$$

which gives (6.1h)–(6.1k). □

6.2 Feasible Region

The next two lemmas characterize the primal part of a KKT solution of (6.1). They state that the primal parts of all optimal solutions differ only in the π values, which together lie on a straight line segment.

Lemma 5. *Let (q, π, Δ) be the primal part of a KKT solution of* (6.1)*. Then the following statements are true.*

1. *$\Delta_v^\pm$ is unique for all nodes $v \in V$.*
2. *q is unique.*
3. *Problem* (6.1) *is convex.*

Proof: Assume that there exist two solutions (q, π, Δ) and (q', π', Δ') of problem (6.1). We define $\hat{q}_e$ for all edges $e \in E$' by $\hat{q}_e := (q'_e - ({\Delta'_e}^+ - {\Delta'_e}^-)) - (q_e - (\Delta_e^+ - \Delta_e^-))$.

Assume there exists a node $v^* \in V$ such that w.l.o.g. $\Delta_{v^*}^+ \neq {\Delta'_{v^*}}^+$ (the case of $\Delta_{v^*}^- \neq {\Delta'_{v^*}}^-$ is similar). W.l.o.g. we can assume that $\Delta_{v^*}^+ - \Delta_{v^*}^- < {\Delta'_{v^*}}^+ - {\Delta'_{v^*}}^-$ (otherwise, exchange the two solutions). Thus we also have that $\Delta_{w^*}^+ - \Delta_{w^*}^- > {\Delta'_{w^*}}^+ - {\Delta'_{w^*}}^-$. We split the flow $\hat{q}$ into a set of path flows $\{P_1, \ldots, P_m\}$ and a set of cycle flows $\{C_1, \ldots, C_n\}$. Let P_ℓ be the path flow that starts at source node v^* and ends at some other sink node w^*. We analyze the node potential differences in the two end nodes of the path P_ℓ, that is, $\pi'_{v^*} - \pi'_{w^*}$ versus $\pi_{v^*} - \pi_{w^*}$, and distinguish three cases.

Case 1, $\Delta_{v^*}^+ - \Delta_{v^*}^- > 0$. Let ε be the amount of flow on path P_ℓ. From constraint (6.1i) for v^* it follows that $\pi_{v^*} = \underline{\pi}_{v^*}$ and $\pi'_{v^*} = \underline{\pi}_{v^*}$. We distinguish three subcases.

Case 1.1, $\Delta_{w^*}^+ - \Delta_{w^*}^- > 0$. Then again it follows that $\pi_{w^*} = \underline{\pi}_{w^*}$. As $\underline{\pi}_{w^*} \leq \pi'_{w^*} \leq \pi_{w^*} = \underline{\pi}_{w^*}$ holds after the augmentation, we obtain $\pi'_{w^*} = \underline{\pi}_{w^*}$. Hence $\pi'_{v^*} - \pi'_{w^*} = \pi_{v^*} - \pi_{w^*}$. Note that $\hat{q}$ represents the difference between flows q and q'. Since the potential differences along path P_ℓ are equal in both cases, the difference in $\hat{q}$ can only affect Δ_e-variables, and not q-variables, because the latter would lead to different node potentials.

Case 1.2, $\Delta_{w^*}^+ - \Delta_{w^*}^- = 0$. Then it follows that $\Delta_{w^*}^+ = \Delta_{w^*}^- = 0$. After augmenting the flow q along P_ℓ we have that ${\Delta'_{w^*}}^+ - {\Delta'_{w^*}}^- < 0$. According to constraint (6.1h) we have that $\pi'_{w^*} = \overline{\pi}_{w^*}$. Now the node potentials are at the lower boundary on the source and at the upper boundary on the sink side. If the augmentation has taken place in q-variables, it would mean that either the node potential at the source is below the lower boundary, or at the sink is above the upper limit, hence infeasible in both cases. Hence again the difference in $\hat{q}$ can only effect Δ_e-variables, and leaves all q-variables unchanged. By the same argument, we have that $\pi_{v^*} = \pi'_{v^*}$ and $\pi_{w^*} = \pi'_{w^*}$.

Case 1.3, $\Delta_{w^*}^+ - \Delta_{w^*}^- < 0$. Then from constraint (6.1h) we have that $\pi'_{w^*} = \overline{\pi}_{w^*}$, and we are in the same situation as in Case 1.2.

Case 2, $\Delta_{v^*}^+ - \Delta_{v^*}^- = 0$. Then it follows that $\Delta_{v^*}^+ = \Delta_{v^*}^- = 0$. From (6.1i) together with $0 = \Delta_{v^*}^+ - \Delta_{v^*}^- < {\Delta'_{v^*}}^+ - {\Delta'_{v^*}}^-$ we get that $\pi'_{v^*} = \underline{\pi}_{v^*}$.

Case 2.1, $\Delta_{w^*}^+ - \Delta_{w^*}^- > 0$. We are in the same situation as in Case 1.1, hence the same conclusion follows.

Case 2.2, $\Delta_{w^*}^+ - \Delta_{w^*}^- = 0$. See Case 1.2.

Case 2.3, $\Delta_{w^*}^+ - \Delta_{w^*}^- < 0$. See Case 1.3.

Case 3, $\Delta_{v^*}^+ - \Delta_{v^*}^- < 0$. Then it follows from (6.1h) that $\pi_{v^*} = \overline{\pi}_{v^*}$.

Case 3.1, $\Delta_{w^*}^+ - \Delta_{w^*}^- > 0$**.** Then it follows from (6.1i) that $\pi_{w^*} = \underline{\pi}_{w^*}$. The argumentation is similar to Case 1.2: If a further augmentation takes place in the q-variables, then either the node potential in v^* would increase, or the node potential in w^* would decrease, what is not possible, since at both ends the node potentials are already at their respective bounds. Hence the augmentation only takes place in Δ_e-variables, and the node potential variables remain unchanged, i.e., $\pi_{v^*} = \pi'_{v^*}$ and $\pi_{w^*} = \pi'_{w^*}$.

Case 3.2, $\Delta_{w^*}^+ - \Delta_{w^*}^- = 0$**.** Then it follows that $\Delta_{w^*}^+ = \Delta_{w^*}^- = 0$. From (6.1i) together with $0 = \Delta_{w^*}^+ - \Delta_{w^*}^- > {\Delta'_{w^*}}^+ - {\Delta'_{w^*}}^-$ we get that $\pi'_{w^*} = \overline{\pi}_{w^*}$. So we are at the upper bound in π_{v^*} (before augmenting) and also at the upper bound in π'_{w^*} (after augmenting along P_ℓ). Assume that the augmentation has affected q-variables. Then the node potential in v^* would be either above the upper limit before augmenting, or the node potential in w^* would be above the upper limit after augmenting along P_ℓ. Hence the augmentation only affected Δ_e-variables, and also the node potentials remain unchanged.

Case 3.3, $\Delta_{w^*}^+ - \Delta_{w^*}^- < 0$**.** Then we get from (6.1i) that $\pi'_{w^*} = \overline{\pi}_{w^*}$. Hence we are in the same situation as in Case 3.2, hence the same conclusions remain valid.

Summing up these 3×3 cases, we have that $\Delta_v^\pm = {\Delta'_v}^\pm$ because any difference leads to a change in the Δ_e variables while the flow variables q_e remain unchanged. This means to increase the objective function value. Thus $\hat{q}$ consists of circulations only. But a circulation can only take place in the Δ_e variables because it would lead to inconsistent potential values otherwise (we note that either q_e changes or $\Delta_e^\pm$ but not both at the same time). Thus $q = q'$. Now that all q-variables do not change, we distinguish two cases. If $\Delta_v^+ > 0$ or $\Delta_v^- > 0$ for some node $v \in V$, then π_v is fixed which implies that π is fixed. The remaining problem of (6.1) is only a linear program in the variables $\Delta = (\Delta_e, \Delta_v)$. In the second case $\Delta_v^+ = 0 = \Delta_v^-$ for all nodes $v \in V$. Again, the remaining problem of (6.1) is only a linear program. This LP is to minimize the objective (6.1a), subject to the flow conservation constraints (6.1c), and some of the Δ_e, Δ_v-variables are fixed to zero by constraints (6.1h)–(6.1k). So it remains a flow on arcs in Δ_e-variables to sources and sinks in Δ_v-variables. This proves, that (6.1) is convex.

□

Lemma 6. *Let* (q, π, Δ) *be the primal part of a KKT solution of* (6.1)*. Then there exist* $a, b \in \mathbb{R}, a \le b$*, and a vector* $\theta \in \mathbb{R}^V \backslash \{0\}$*, such that* $(q, \hat{\pi}, \Delta)$ *is also primal part of some KKT solution of* (6.1)*, for* $\hat{\pi} := \pi + t\theta$ *and all* $t \in [a, b]$*. That is, all possible* π *vectors lie on a segment of a straight line.*

Proof: We know from the previous Lemma 5 that q and $\Delta_v^\pm$ are unique. Hence problem (6.1a) reduces to constraint (6.1b) and the bound constraints (6.1d), (6.1e), and (6.1h), (6.1i) for those $v \in V$ with $\Delta_v^+ > 0$ or $\Delta_v^- > 0$ when ignoring $\Delta_e^\pm$. If we relax these four bound constraints, we are in the same situation as in Theorem 1. Thus, there exists a $\theta \in \mathbb{R}^V \backslash \{0\}$, such that $(q, \hat{\pi})$ is a feasible solution for constraint (6.1b), for all $t \in \mathbb{R}$ and $\hat{\pi} := \pi + t\theta$. Now taking the bound constraints (6.1d), (6.1e), (6.1h), and (6.1i) into account, the range $t \in \mathbb{R}$ is restricted to an interval by these bounds. Hence t can only vary in such way that $\underline{\pi}_v \le \hat{\pi} = \pi + t\theta \le \overline{\pi}_v$ remains valid. This restricts t to the interval $[a, b]$, where

$$a := \max\left\{\frac{\pi_v - \underline{\pi}_v}{\theta_v} \mid v \in V, \theta_v \neq 0\right\}, \qquad b := \max\left\{\frac{\overline{\pi}_v - \pi_v}{\theta_v} \mid v \in V, \theta_v \neq 0\right\}.$$

□

From Lemma 5 and Lemma 6 the following result follows.

Theorem 5. *A local optimal KKT solution to problem* (6.1) *is already a global optimal one.*

Because of Theorem 5 we can use a local NLP solver (IPOPT, for instance) to solve the relaxed leaf problem (6.1) to global optimality.

We note that we can solve problem (6.1) without any bounds on the flow variables as the solution is unique by Lemma 5. If it is feasible we get the solution and verify the bounds afterwards. If we do not obtain a feasible solution, or if we obtain a feasible solution that lies outside the bounds, then problem (4.3) is infeasible.

6.3 Interpretation of Lagrange Dual Variables

For some Lagrange parameters $(\mu, \lambda) = (\mu_v, \mu_e, \mu_v^+, \mu_v^-, \mu_e^+, \mu_e^-, \lambda_v^+, \lambda_v^-, \lambda_e^+, \lambda_e^-, \tilde{\lambda}_v^+, \tilde{\lambda}_v^-, \tilde{\lambda}_e^+, \tilde{\lambda}_e^-)$, such that $\mu_v, \mu_e, \mu_v^+, \mu_v^-, \mu_e^+, \mu_e^- \in \mathbb{R}$ and $\lambda_v^+, \lambda_v^-, \lambda_e^+, \lambda_e^-, \tilde{\lambda}_v^+, \tilde{\lambda}_v^-, \tilde{\lambda}_e^+, \tilde{\lambda}_e^- \in \mathbb{R}_{\geq 0}$, the Lagrangian of problem (6.1) has the form

$$
\begin{aligned}
L(q, \pi, \Delta, \mu, \lambda) = &\sum_{v \in V} \left(\Delta_v^+ + \Delta_v^-\right) + \sum_{e \in E'} \left(\Delta_e^+ + \Delta_e^-\right) \\
&+ \sum_{e=(v,w) \in E'} \mu_e \left(\Phi(q_e) - (\pi_v - \gamma_e \pi_w)\right) \\
&+ \sum_{v \in V} \mu_v \left(s_v - \sum_{e \in \delta^+(v)} (q_e - \Delta_e^+ + \Delta_e^-) + \sum_{e \in \delta^-(v)} (q_e - \Delta_e^+ + \Delta_e^-) + (\Delta_v^+ - \Delta_v^-) \right) \\
&+ \sum_{v \in V} \left(\lambda_v^+ (\pi_v - \overline{\pi}_v) + \lambda_v^- (\underline{\pi}_v - \pi_v)\right) \\
&+ \sum_{e \in E'} \left(\lambda_e^+ (q_e - \overline{q}_e) + \lambda_e^- (\underline{q}_e - q_e)\right) \\
&+ \sum_{v \in V} \left(\mu_v^+ (\overline{\pi}_v - \pi_v) \Delta_v^- + \mu_v^- (\pi_v - \underline{\pi}_v) \Delta_v^+\right) \\
&+ \sum_{e \in E'} \left(\mu_e^+ (\overline{q}_e - q_e) \Delta_e^- + \mu_e^- (q_e - \underline{q}_e) \Delta_e^+\right) \\
&- \sum_{v \in V} \left(\tilde{\lambda}_v^+ \Delta_v^+ + \tilde{\lambda}_v^- \Delta_v^-\right) \\
&- \sum_{e \in E'} \left(\tilde{\lambda}_e^+ \Delta_e^+ + \tilde{\lambda}_e^- \Delta_e^-\right).
\end{aligned}
\tag{6.6}
$$

For a local optimum of a nonlinear problem it is shown by Boyd and Vandenberghe [4] that there exist values for these dual variables fulfilling the KKT conditions. From these KKT conditions we derive the following constraints:

$$
\begin{aligned}
&\frac{\partial L}{\partial q_e}: && \mu_e \left(\nabla_{q_e} \Phi_e(q_e)\right) + \lambda_e^+ - \lambda_e^- = \mu_v - \mu_w && \forall e = (v, w) \in E', && (6.7a) \\
&\frac{\partial L}{\partial \pi_v}: && \sum_{e \in \delta^+(v)} \mu_e - \sum_{e \in \delta^-(v)} \gamma_e \mu_e = \lambda_v^+ + \mu_v^+ \Delta_v^- - (\lambda_v^- + \mu_v^- \Delta_v^+) && \forall v \in V, && (6.7b) \\
&\frac{\partial L}{\partial \Delta_v^+}: && \mu_v + \mu_v^- (\pi_v - \underline{\pi}_v) - \tilde{\lambda}_v^+ = -1 && \forall v \in V, && (6.7c) \\
&\frac{\partial L}{\partial \Delta_v^-}: && -\mu_v + \mu_v^+ (\overline{\pi}_v - \pi_v) - \tilde{\lambda}_v^- = -1 && \forall v \in V, && (6.7d) \\
&\frac{\partial L}{\partial \Delta_e^+}: && \mu_v - \mu_w + \mu_e^- (q_e - \underline{q}_e) - \tilde{\lambda}_e^+ = -1 && \forall e \in E', && (6.7e) \\
&\frac{\partial L}{\partial \Delta_e^-}: && \mu_w - \mu_v + \mu_e^+ (\overline{q}_e - q_e) - \tilde{\lambda}_e^- = -1 && \forall e \in E'. && (6.7f)
\end{aligned}
$$

From this set of constraints we can derive the following conditions:

$$
\begin{aligned}
&\Delta_v^- > 0 \Rightarrow \mu_v = 1, && \Delta_v^+ > 0 \Rightarrow \mu_v = -1, && \forall\, v \in V, && (6.8a) \\
&\Delta_e^- > 0 \Rightarrow \mu_v - \mu_w = 1, && \Delta_e^+ > 0 \Rightarrow \mu_v - \mu_w = -1, && \forall\, e = (v, w) \in E', && (6.8b) \\
&\pi_v < \overline{\pi}_v \Rightarrow \lambda_v^+ + \mu_v^+ \Delta_v^- = 0, && \pi_v > \underline{\pi}_v \Rightarrow \lambda_v^- + \mu_v^- \Delta_v^+ = 0, && \forall\, v \in V, && (6.8c) \\
&q_e < \overline{q}_e \Rightarrow \lambda_e^+ = 0, && q_e > \underline{q}_e \Rightarrow \lambda_e^- = 0 && \forall\, e \in E'. && (6.8d)
\end{aligned}
$$

We compare (6.7) with (5.18). Most of the interpretation after (5.18) remain valid also for (6.7). In the following, we focus on the differences. Constraints (6.7a) correspond to (5.18a), and constraints (6.7b) correspond to (5.18b). The derived conditions (5.19a) and (5.19b) state that under certain cases for the node potential values we have to fix the dual node flows. Complementary, the derived conditions (6.8a) state that under certain cases for the node flow slack values we have to fix the dual node potentials. Similar, the derived conditions (5.19c) and (5.19d) state that under certain cases for the arc flow values we have to fix the dual variables λ_e^+ and λ_e^-. Complementary, the derived conditions (6.8b) state that under certain cases for the arc flow slack values we have to fix the dual node potential difference.

If we consider problem (6.1) without bounds on the arc flows, that is, without constraints (6.1f) and (6.1g) (thus, also without constraints (6.1j) and (6.1k)), and compute a KKT solution, then the dual flow μ in this solution does not have a cycle flow. (A cycle flow would contradict (6.7a).) Consider an augmenting path from some $v \in V$ with $\Delta_v^- > 0$ to some other $w \in V, v \neq w$ with $\Delta_w^+ > 0$. Then $\mu_v = 1$ and $\mu_w = -1$, hence there must be an arc $e' = (v', w')$ in the path with $\mu_{v'} - \mu_{w'} > 0$. Then from (6.7a) it follows that $\mu_{e'} > 0$ (because $\lambda_{e'}^{\pm} = 0$ and the gradient is nonnegative). Since we do not have a cycle flow, this arc belongs to another path from v_0 to w_0, where $\pi_{v_0} = \overline{\pi}_{v_0}$ and $\pi_{w_0} = \underline{\pi}_{w_0}$. Hence we cannot send more flow over arc e'. Since these arguments hold for any potentially augmenting path, the respective arcs e' form (a kind of) a cut that prevents from sending more flow between sources and sinks.

7 Relaxation of Potential-Flow-Coupling Constraints

As third possibility we consider the following relaxation to compute a solution to the leaf problem (4.3):

$$\min \sum_{e \in E'} \Delta_e^+ + \Delta_e^- \quad \text{s. t.} \tag{7.1a}$$

$$\Phi_e(q_e) - (\pi_v - \gamma_e \pi_w) - (\Delta_e^+ - \Delta_e^-) = 0 \qquad \forall e \in E', \tag{7.1b}$$

$$\sum_{e \in \delta^+(v)} q_e - \sum_{e \in \delta^-(v)} q_e = s_v \qquad \forall v \in V, \tag{7.1c}$$

$$\pi_v - \overline{\pi}_v \le 0 \qquad \forall v \in V, \tag{7.1d}$$

$$\underline{\pi}_v - \pi_v \le 0 \qquad \forall v \in V, \tag{7.1e}$$

$$q_e - \overline{q}_e \le 0 \qquad \forall e \in E', \tag{7.1f}$$

$$\underline{q}_e - q_e \le 0 \qquad \forall e \in E', \tag{7.1g}$$

$$\pi_v \in \mathbb{R} \qquad \forall\, v \in V, \tag{7.1h}$$

$$q_e \in \mathbb{R} \qquad \forall\, e \in E', \tag{7.1i}$$

$$\Delta_e^{\pm} \in \mathbb{R}_+ \qquad \forall\, e \in E'. \tag{7.1j}$$

Basically, this is a relaxation of the potential-flow-coupling constraints (4.3a). If the nonlinear optimization problem (7.1) is feasible and (q^*, π^*, Δ^*) denotes its optimal solution, then (q^*, π^*) is feasible for the leaf problem (4.3), if and only if the optimal objective value equals zero, i.e., $(\Delta_e^{\pm})^* = 0$ for all $e \in E'$.

In the following we will show, that this problem is a non-convex optimization problem having different local optimal solutions. Hence our techniques developed above cannot be adopted to this type of relaxation.

7.1 Solving KKT Systems

For some Lagrange parameters $(\mu, \lambda) = (\mu_v, \mu_e, \lambda_v^+, \lambda_v^-, \lambda_e^+, \lambda_e^-, \tilde{\lambda}_e^+, \tilde{\lambda}_e^-)$, such that $\mu_v, \mu_e \in \mathbb{R}$ and $\lambda_v^+, \lambda_v^-, \lambda_e^+, \lambda_e^-, \tilde{\lambda}_e^+, \tilde{\lambda}_e^- \in \mathbb{R}_+$, the Lagrangian of problem (7.1) has the form

$$L(q, \pi, \Delta, \mu, \lambda) = \sum_{e \in E'} \left(\Delta_e^+ + \Delta_e^-\right) \tag{7.2}$$

$$+ \sum_{e \in E'} \mu_e \left(\Phi_e(q_e) - (\pi_v - \gamma_e \pi_w) - (\Delta_e^+ - \Delta_e^-) \right) \tag{7.3}$$

$$+ \sum_{v \in V} \mu_v \left(s_v - \sum_{e \in \delta^+(v)} q_e + \sum_{e \in \delta^-(v)} q_e \right) \tag{7.4}$$

$$+ \sum_{v \in V} \left(\lambda_v^+ (\pi_v - \overline{\pi}_v) + \lambda_v^- (\underline{\pi}_v - \pi_v) \right) \tag{7.5}$$

$$+ \sum_{e \in E'} \left(\lambda_e^+ (q_e - \overline{q}_e) + \lambda_e^- (\underline{q}_e - q_e) \right) \tag{7.6}$$

$$- \sum_{e \in E'} \tilde{\lambda}_e^+ \Delta_e^+ + \tilde{\lambda}_e^- \Delta_e^- . \tag{7.7}$$

For a local optimum of a nonlinear problem it is shown by Boyd and Vandenberghe [4] that there exist values for these dual variables fulfilling the KKT conditions. From these KKT conditions we derive the following constraints:

$$\frac{\partial L}{\partial q_e} : \quad \mu_e \left(\nabla_{q_e} \Phi_e(q_e) \right) + \lambda_e^+ - \lambda_e^- = \mu_v - \mu_w \qquad \forall e = (v, w) \in E', \tag{7.8a}$$

$$\frac{\partial L}{\partial \pi_v} : \quad \sum_{e \in \delta^+(v)} \mu_e - \sum_{e \in \delta^-(v)} \gamma_e \mu_e = \lambda_v^+ - \lambda_v^- \qquad \forall v \in V, \tag{7.8b}$$

$$\frac{\partial L}{\partial \Delta_e^+} : \quad \mu_e + \tilde{\lambda}_e^+ = 1 \qquad \forall e \in E', \tag{7.8c}$$

$$\frac{\partial L}{\partial \Delta_e^-} : \quad \mu_e - \tilde{\lambda}_e^- = -1 \qquad \forall e \in E'. \tag{7.8d}$$

We derive from the complementarity constraints:

$$q_e < \overline{q}_e \Rightarrow \lambda_e^+ = 0, \quad q_e > \underline{q}_e \Rightarrow \lambda_e^- = 0 \qquad \forall\, e \in E', \tag{7.9a}$$

$$\pi_v < \overline{\pi}_v \Rightarrow \lambda_v^+ = 0, \quad \pi_v = \overline{\pi}_v \Rightarrow \lambda_v^+ \geq 0 \qquad \forall v \in V, \tag{7.9b}$$

$$\pi_v > \underline{\pi}_v \Rightarrow \lambda_v^- = 0, \quad \pi_v = \underline{\pi}_v \Rightarrow \lambda_v^- \leq 0 \qquad \forall v \in V, \tag{7.9c}$$

$$\Delta_e^+ > 0 \Rightarrow \mu_e = 1, \quad \Delta_e^- > 0 \Rightarrow \mu_e = -1 \qquad \forall e \in E'. \tag{7.9d}$$

We compare (7.8) with (6.7) and (5.18). Remember that in the first relaxation, (5.18), we had an enforcement of the variables $\lambda_v^\pm, \lambda_e^\pm$. In the second relaxation, (6.7), we derived an enforcement of the variables μ_v. Now in the third relaxation, (7.8), only an enforcement of the variables μ_e to nonzero values remains.

7.2 Feasible Region

The following example shows that the feasible domain of (7.1) is non-convex in general.

Example 1. Consider the network of two nodes v, w and two parallel arcs e_1, e_2 from v to w. We set $\Phi_e(q_e) := \alpha_e\, q_e |q_e|$ and assume arc constants $\alpha_1 = 1.0$ and $\alpha_2 = 1.5$, and node potential bounds $\underline{\pi}_v = -10, \overline{\pi}_v = 0, \underline{\pi}_w = 10, \overline{\pi}_w = 20$. Node v is an entry with flow $+1$, node w is an exit with flow -1. Then the following two solutions both fulfill the KKT system (7.8):

- Let $q_1 = 3, q_2 = -2, \pi_v = 0, \pi_w = 10, \Delta_1^+ = 19, \Delta_2^+ = 4$ (and zero otherwise). The objective function value is 23. The dual values are $\mu_1 = 1, \mu_2 = 1, \mu_v = 6$.
- Let $q_1 = 0.6, q_2 = 0.4, \pi_v = 0, \pi_w = 10, \Delta_1^+ = 10.36, \Delta_2^+ = 10.24$ (and zero otherwise). The objective function value is 20.6. The dual values are $\mu_1 = 1, \mu_2 = 1, \mu_v = 1.2$.

Thus we found two different solutions, both fulfilling the KKT conditions. Hence they are local optimal, which shows that the feasible domain is non-convex.

8 Computational Results

Osiadacz et al. [24] give an iterative procedure to compute a solution to the leaf problem. They consider the following problem:

$$\begin{aligned} \Phi(q_e) - (\pi_v - \pi_w) &= 0 \qquad (e = (v,w) \in E') \\ \sum_{e \in \delta^+(v)} q_e - \sum_{e \in \delta^-(v)} q_e - s_v &= 0 \qquad (v \in V). \end{aligned} \tag{8.1}$$

Writing the second constraints as $Aq = s$, where A denotes the graph adjacency matrix, we seek a vector π, such that

$$A\Phi^{-1}(A^T\pi) - s = 0.$$

They propose to solve this nonlinear equation system by the Newton method and give computational results. Furthermore they note, that solving this system by the Newton method is equivalent to adapting the potential values π iteratively (this would correspond to a special newton step). We did not re-implement their combinatorial algorithm, because it only means that special Newton steps are applied to the system (8.1). Instead, we use a general-purpose solver code that also uses Newton steps, but based on more elaborate selection criteria. Moreover, the use of a modern NLP solver (such as IPOPT) allows to include a presolving before actually applying the Newton steps. This combination is necessary when dealing with large-scale instances. Our approach is to do presolving, then solve the LP relaxation, and finally solve the remaining NLP.

As test instances we use five different networks. The first network of practical dimension (`net1a`) is shown in Figure 5.3. This network contains active elements (compressors and control valves) that we do not discretize (i.e., y_e is a continuous variable). The implication of this is that we cannot decide on infeasibility with our relaxation strategies – only the spatial branching approach can determine infeasibility. However, this case does not come up in our test instances. Network `net1a` is publicly available at URL `http://gaslib.zib.de` under the name `gaslib-582`. We further used two variants of this network, called `net1b` and `net1c`, which have slightly different pipes and active elements. These networks are in industrial use, but to large extend they are similar to the public network `net1a`. The final two networks (`net2`, `net3`) are smaller test networks, see Figure 8.1 and Figure 8.2. These two networks do not carry active elements for pressure regulation (compressors and control valves), but valves for opening or closing (new) pipelines. The dimensions of the underlying graphs are summarized in Table 1.

instance	nodes	pipes	active elements
`net1a`	582	451	62
`net1b`	661	498	75
`net1c`	592	452	72
`net2`	135	140	103
`net3`	367	402	261

Table 1. The sizes of the five test instances.

We implemented the algorithms described above in C on a cluster of 64bit Intel Xeon X5672 CPUs at 3.20 GHz with 12 MByte cache and 48 GB main memory, running an OpenSuse 12.1 Linux with a gcc 4.6.2 compiler. We used the following software packages: SCIP 3.0.1 as mixed-integer nonlinear branch-and-cut framework (for details on SCIP we refer to [1]), CPLEX 12.1 [20] as linear programming solver, Ipopt 3.10 [30] as nonlinear solver, and Lamatto++ [17] as framework for handling the input data. Hyperthreading and Turboboost were disabled. In all experiments, we ran only one job per node to avoid random noise in the measured running time that might be caused by cache-misses if multiple processes share common resources.

We compare four strategies for solving the subproblems (leaf problems) (4.3). Heuristics of SCIP were disabled in order to get a more accurate comparison of the different strategies. The first strategy is to use plain SCIP. All branching decisions are up to the solver, and the nonlinear subproblems are only solved by cutting planes, spacial branching, and the linear solver CPLEX. The second strategy is to enforce a certain branching priority rule, so that SCIP first branches on binary decision variables (x_e), and only

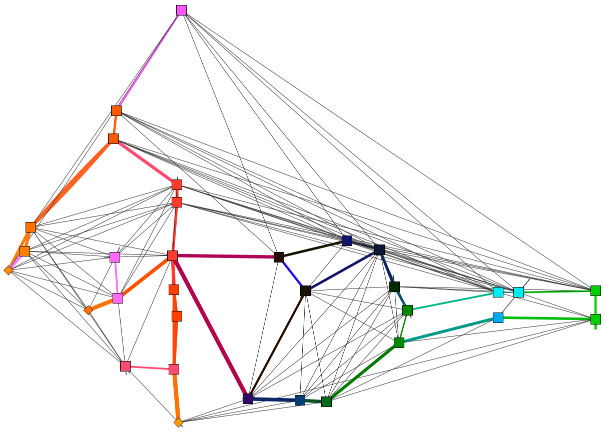


Fig. 8.1. The test network `net2`. Coloured arcs form the original network, while gray arcs are extension pipes.

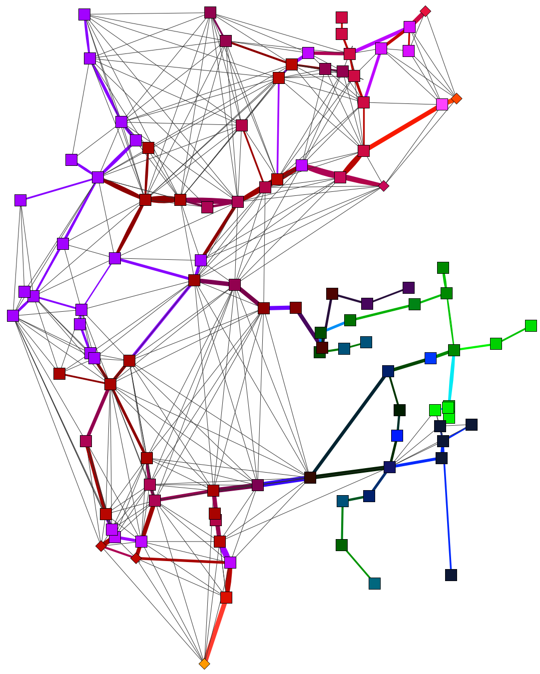


Fig. 8.2. The test network `net3`. All coloured arcs form the original network, while extension pipes are colored in gray.

after all binaries are fixed, it is allowed to perform spatial branching on continuous variables. In the third and forth strategy we replace the linear solver CPLEX by the nonlinear solver IPOPT. The third strategy implements the domain relaxation from Section 5, and the forth strategy uses the relaxation of the flow conservation constraints from Section 6. We did not implement the method described in Section 7, because the resulting subproblems are nonconvex, and thus a local solver cannot guarantee to find a global optimum of the leaf problem, which is necessary to prune the node in the branch-and-bound process.

We start with network **net2** and **net3**. Here we solve a topology optimization problem. That is, we add an objective function of the form

$$\sum_{e \in E_a} c_e x_e \quad \rightarrow \quad \min \tag{8.2}$$

to problem (4.2). This objective function models the cost for constructing new pipelines or other active elements (compressors or control valves) to the network, and the whole optimization problem aims for identifying a minimum cost extension to the network, such that an otherwise infeasible flow becomes feasible. For further details on topology optimization using the here presented model, and related algorithmic aspects, we refer to [16]. Figure 8.3 summarizes the runtime results for the three strategies on the test instances **net2** and **net3**. We solved 8 instances altogether, where 5 different nominations were solved for **net2** and 3 for **net3**. The three graphs show the share of instances (in percent) that could be solved within a certain time limit. The graph for spatial branching is below the graph for the flow conservation relaxation, which is below the graph for the domain relaxation. Here we see a clear ordering of these three strategies. The detailed results that are underlying these plots can be found in Table 2. In each of the 5 runs of **net2**, altogether 1556 leaf problems were solved with one of the two relaxations. For **net3**, altogether 53080 leaf problems were solved. Here we do not count those instances that were detected as infeasible during presolve, only those that are really solved by an NLP solver. This number corresponds to the number of crosses in Figure 8.4. This figure shows a scatter plot of the runtime for the NLP solver (without presolving time). One can see that the runtime for the domain relaxation is one order of magnitude lower than for the flow conservation relaxation.

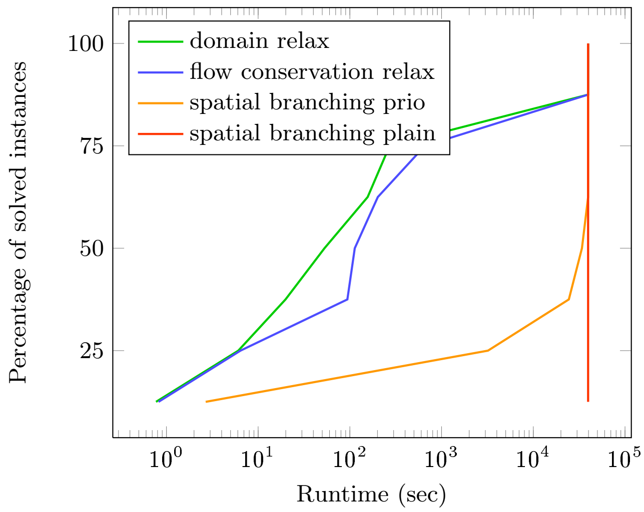


Fig. 8.3. Runtime for instances **net2** and **net3** (aggregated).

For network **net1a** we omit an objective function and solve for feasibility. Among them are 30 nominations from our industrial partner, and additionally 170 randomly generated nominations. Figure 8.5 shows the percentage of instances that could be solved within a given time limit. In general, the domain relaxation outperforms the flow conservation relaxation and the spatial branching strategy. The spatial branching strategy is better than the flow conservation relaxation for a tight time limit (less than 2 seconds), but for larger time limits (more than 2 seconds) the flow conservation relaxation becomes

	spatial branching plain				spatial branching prio				domain relaxation				flow cons. relaxation			
instance	gap	primal	time	nodes	gap	primal	time	nodes	gap	primal	time	nodes	gap	primal	time	nodes
`net2.1`	n/a	n/a	limit	3644521	0.0	0.00	2.68	1276	0.0	0.00	0.77	104	0.0	0.00	0.82	104
`net2.2`	n/a	n/a	limit	5984307	0.0	95.00	3206.86	2236848	0.0	95.00	6.08	105	0.0	95.00	6.42	105
`net2.3`	330.71	2263.19	limit	4856305	n/a	n/a	24403.18	15292006	0.0	525.46	19.88	1735	0.0	525.46	93.94	1735
`net2.4`	n/a	n/a	limit	4983712	n/a	n/a	limit	8503282	0.0	997.54	52.88	2428	0.0	997.54	113.11	2428
`net2.5`	n/a	n/a	limit	3799508	n/a	n/a	limit	7463623	0.0	1545.78	271.26	39778	0.0	1545.78	657.68	39126
`net3.1`	n/a	n/a	limit	636165	n/a	n/a	33916.44	6664665	0.0	555.74	156.36	1700	0.0	555.74	201.74	1700
`net3.2`	n/a	n/a	limit	1545581	n/a	n/a	limit	3958916	93.24	2578.95	limit	375249	99.9	2578.95	limit	219920
`net3.3`	n/a	n/a	limit	899928	n/a	n/a	limit	3808846	204.89	7341.47	limit	2124709	n/a	n/a	limit	1838273

Table 2. Results for topology optimization for `net2` and `net3`.

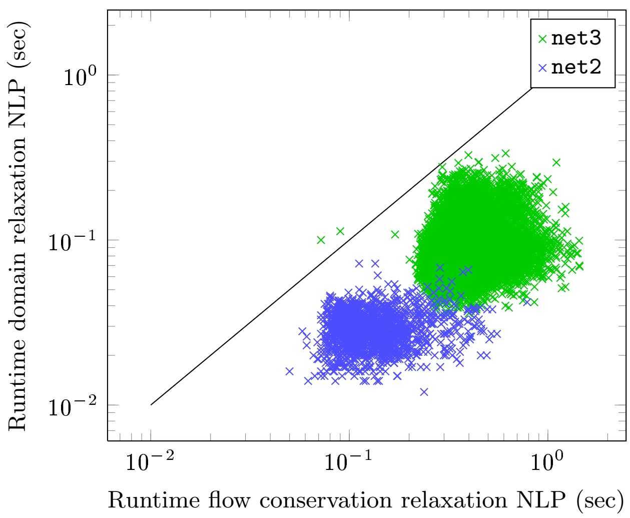


Fig. 8.4. Runtime comparison for the two relaxation strategies on instance `net2` and `net3`.

better. For very short runtimes (less than 1 second) there is no clear winner among the three strategies. We compare the two relaxation strategies in detail in the scatter plot in Figure 8.6. Again, the domain relaxation is faster than the flow conservation relaxation by one order of magnitude.

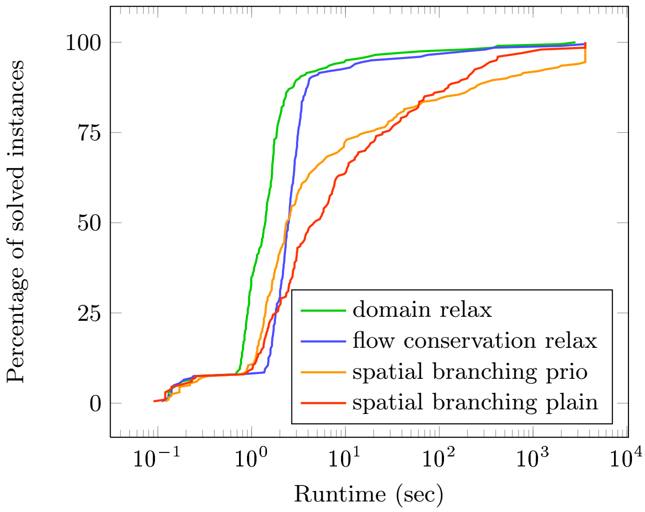


Fig. 8.5. Runtime for instances `net1a`, `net1b` and `net1c`.

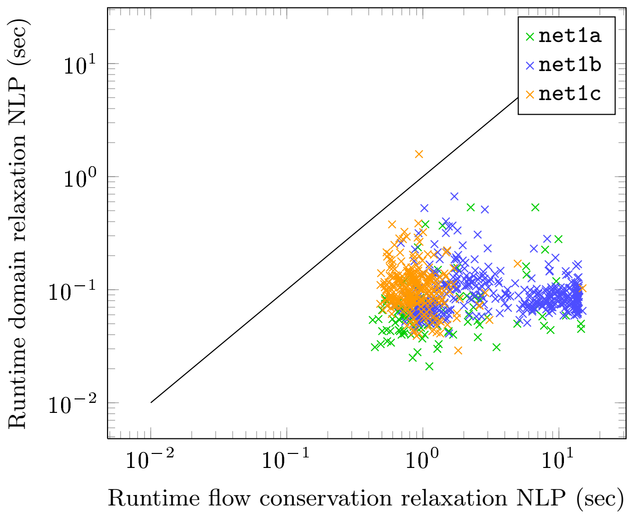


Fig. 8.6. Runtime comparison for the two relaxation strategies on instance `net1a`, `net1b` and `net1c`. The index above the line had numerical troubles and ran into the timelimit.

9 Conclusions

We presented different relaxation strategies for nonlinear network flow problems with node potentials. We gave iterative solution methods and computationally demonstrated that the use of local nonlinear solvers outperforms spatial branching methods. We determined that the domain relaxation method is about a factor of 10 faster than the flow conservation relaxation method. In our practical applications, we therefore use the domain relaxation method as a subroutine.

Still, the flow conservation relaxation method has its practical merits: In a certain practical application case of the method one does not specify a single nomination vector $(s_v)_{v \in V}$, but interval data $[\underline{s}_v, \overline{s}_v]$ for each node $v \in V$. Such "interval nomination" is considered as feasible, as soon as there *exists* one $s_v \in [\underline{s}_v, \overline{s}_v]$ for each $v \in V$, such that the nomination with these values is feasible. In the model, one replaced the equality constraint (4.2g) by two inequality constraints with the interval as lower and upper bounds. This question can be addressed with the spatial branching approach and also with the flow conservation relaxation (and we showed that the latter is the faster of these two), but not with the domain relaxation method. One can easily adapt the constraints (4.3b) and (6.1c) to handle interval data, but there exists a counterexample for the domain relaxation model (5.1), i.e., a KKT solution that violates the pressure bounds.

Acknowledgements

We are grateful to Open Grid Europe GmbH (OGE, Essen/Germany) for supporting our work. The first coauthor conducted parts of this research under a Konrad-Zuse-Fellowship.

References

1. T. Achterberg. SCIP: Solving Constraint Integer Programs. *Mathematical Programming Computation*, 1(1):1–41, 2009.
2. Frédéric Babonneau, Yurii Nesterov, and Jean-Philippe Vial. Design and operations of gas transmission networks. *Operations Research*, 60(1):34–47, 2012.
3. Timo Berthold, Stefan Heinz, and Stefan Vigerske. Extending a CIP framework to solve MIQCPs. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154, part 6 of *The IMA Volumes in Mathematics and its Applications*, pages 427–444. Springer, 2012. Also available as ZIB-Report 09-23.
4. S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2004.
5. M. Collins, L. Cooper, R. Helgason, J. Kennington, and L. LeBlanc. Solving the pipe network analysis problem using optimization techniques. *Management science*, 24(7):747–760, 1978.
6. A.R. Conn, N.I.M. Gould, and P.L. Toint. *Trust-Region Methods.* SIAM, Philadelphia, PA, USA, 2000.
7. D. De Wolf. Mathematical properties of formulations of the gas transmission problem. Online available at URL http://www-heb.univ-littoral.fr/dewolf, April 2004. Submitted to RAIRO Operations Research.
8. D. De Wolf and B. Bakhouya. The gas transmission problem when the merchant and the transport functions are disconnected. Technical Report 01/01, Ieseg, Université catholique de Lille, HEC Ecole de Gestion de l'ULG, January 2007.
9. D. De Wolf and B. Bakhouya. Optimal dimensioning of pipe networks: the new situation when the distribution and the transportation functions are disconnected. Technical Report 07/02, Ieseg, Université catholique de Lille, HEC Ecole de Gestion de l'ULG, July 2008.
10. D. De Wolf and B. Bakhouya. Solving gas transmission problems by taking compressors into account. Online available at URL http://www-heb.univ-littoral.fr/dewolf, September 2008. Submitted to 4OR.
11. D. de Wolf and Y. Smeers. Optimal dimensioning of pipe networks with application to gas transmission networks. *Operations Research*, 44(4):596–608, July-August 1996.
12. D. De Wolf and Y. Smeers. The gas transmission problem solved by an extension of the simplex algorithm. *Management Science*, 46(11):1454–1465, November 2000.
13. Ron S. Dembo, John M. Mulvey, and Stavros A. Zenios. Large-scale nonlinear network models and their application. *Operations Research*, 37(3):pp. 353–372, 1989.
14. L.R. Ford and D.R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399 – 404, 1956.
15. A. Fügenschuh, H. Homfeld, H. Schülldorf, and S. Vigerske. Mixed-integer nonlinear problems in transportation applications. In H. Rodrigues et al., editor, *Proceedings of the 2nd International Conference on Engineering Optimization (CD-ROM)*, 2010.
16. Armin Fügenschuh and Jesco Humpola. A new class of valid inequalities for nonlinear network design problems. Technical Report 13-06, ZIB, Takustr.7, 14195 Berlin, 2013.
17. B. Geißler, A. Martin, and A. Morsi. Lamatto++. information available at URL `http://www.mso.math.fau.de/edom/projects/lamatto.html`, February 2013.
18. I. Gentilini, F. Margot, and K. Shimada. The Traveling Salesman Problem with Neighborhoods: MINLP Solution. *Optimization Methods and Software*, 28:364 – 378, 2013.
19. A. Hazen and G. S. Williams. *Hydraulic Tables.* John Wiley and Sons, New York, 1920.
20. IBM Corporation, Armonk, USA. *User's Manual for CPLEX*, 12.1 edition, 2011.

21. J. J. Maugis. Etude de réseaux de transport et de distribution de fluide. *RAIRO Operations Research*, 11(2):243–248, 1977.
22. G. L. Nemhauser and L. A. Wolsey. Integer programming. In G. L. Nemhauser, A. H. G. Rinnooy Kan, and M. J. Todd, editors, *Optimization*, chapter 6, pages 447–527. Elsevier, 1989.
23. Jeffrey Oldham. Combinatorial approximation algorithms for generalized flow problems. In *Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms SODA'99*, pages 704 – 714, 1999.
24. Andrzej J. Osiadacz. *Simulation and analysis of gas networks.* Gulf Publishing Company, 1987.
25. 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. Technical report, ZIB-Report 12-41, Zuse Institute Berlin, Takustr.7, 14195 Berlin, 2012.
26. H. D. Sherali and E. P. Smith. An optimal replacement-design model for a reliable water distribution network system. In Bryan Coulbeck, editor, *Integrated computer applications in water supply (vol. 1)*, pages 61–75. John Wiley & Sons, Inc., New York, NY, USA, 1994.
27. E.M.B. Smith and C.C. Pantelides. A symbolic reformulation/spatial branch-and-bound algorithm for the global optimization of nonconvex MINLPs. *Computers & Chemical Engineering*, 23:457–478, 1999.
28. M. Tawarmalani and N.V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.
29. M. Tawarmalani and N.V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.
30. A. Wächter and L. T. Biegler. On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1):25–57, 2006.
31. T. R. Weymouth. Problems in Natural Gas Engineering. *Transactions of the American Society of Mechanical Engineers*, 34:185–231, 1912.