A partial outer convexification approach to control transmission lines

S. Göttlich*, A. Potschka†, C. Teuber*

November 2, 2017

Abstract

In this paper we derive an efficient optimization approach to calculate optimal controls of electric transmission lines. These controls consist of time-dependent inflows and switches that temporarily disable single arcs or whole subgrids to reallocate the flow inside the system. The aim is then to find the best energy input in terms of boundary controls in combination with the optimal configuration of switches, where the dynamics is driven by a coupled system of hyperbolic differential equations. Our optimization approach is a two-step heuristic based on the idea of partial outer convexification. We examine the applicability of a discrete approximation lemma and introduce a third step to improve the quality of the heuristic. A comparison with a direct solver yields very promising results.

AMS subject classifications: 35L65, 49J20, 90C11, 90C35
Keywords: transmission lines, optimization, partial outer convexification

1 Introduction

In a world full of electrical devices, energy management has become one of the most relevant fields of research. Various publications focus on different aspects of energy supply [2, 16, 18, 30, 38, 45, 46, 47, 54]. This work will be based on [27]. Therein the computation of optimal power source outputs for a given set of customer demands in a power network has been considered.

In current research [20, 41, 44] the focus lies on the opportunity to temporarily change the topology of the network and thus on the opportunity to influence the distribution at inner vertices. The so-called optimal transmission switching is based on the classic power flow problem [7, 9, 14, 37], an algebraic system of nonlinear equations. We will contribute to this by deriving an alternative approach based on the telegraph equations [22, 25, 34], which are a system of hyperbolic partial differential equations (PDEs). We will present a model which is an extension of the power network problem from [27], which contains additional binary variables. This results in a mixed-integer nonlinear optimization problem (MINLP), which requires more sophisticated solution methods that exploit the structure of the problem.

Therefore, an alternative approach, partial outer convexification, has reached popularity in the field of optimal control problems constrained by ordinary differential equations. In outer convexification [48, 49], a heuristic has been derived, that approximates the optimal solution of the MINLP by first solving the NLP-relaxation and afterwards rounding the values of the binary variables via a suitable rounding strategy given by a mixed-integer linear problem (MILP). This approach has rarely been applied to PDE-constrained problems [26, 32, 33]. Outer convexification has the advantage of an approximation lemma that guarantees the convergence the approximate solution against the solution of the NLP-relaxation if the time step size approaches zero for a fixed spatial step size. In general, this result should be handled with care. Considering the time and

*University of Mannheim, Department of Mathematics, 68131 Mannheim, Germany (goettlich@uni-mannheim.de, cteuber@mail.uni-mannheim.de).
†Heidelberg University, Interdisciplinary Center for Scientific Computing, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany (potschka@iwr.uni-heidelberg.de).

space step sizes individually can have a severe effect on the approximation of the partial differential equation due to effects like numerical diffusion.

The outline is as follows: In section 2 we extend the optimization problem from [27] to the MINLP of interest. Then we present a valid discretization and additional constraints that occur in practice. In section 3 we introduce the two-step heuristic of partial outer convexification and prove that an important approximation result can be applied in the case of the extended power network problem. Furthermore, we extend the heuristic by a third step, which involves the solution of an additional NLP, which can be viewed as a postprocessing step to obtain more reliable solutions. We conclude this article in section 4 by comparing the heuristic with directly solving the MINLP.

2 Modeling

Several different mathematical models for transmission lines or power grids are present in the literature. Nevertheless, our starting point is the model considered in [25, 27], which is based on the telegraph equations, a coupled 2×2-system of balance laws describing the transport along a single transmission line with space coordinate $x \in \mathbb{R}$ and time coordinate $t \in \mathbb{R}^+$

$$\partial_t \xi(x,t) + \Lambda \, \partial_x \xi(x,t) + B \, \xi(x,t) = 0, \tag{1}$$

where

$$\Lambda = \begin{pmatrix} \lambda^+ & 0 \\ 0 & \lambda^- \end{pmatrix}, \text{ with } \lambda^\pm = \pm(\sqrt{LC})^{-1}$$

and

$$B = [b_{ij}] = \frac{1}{2} \begin{pmatrix} RL^{-1} + GC^{-1} & RL^{-1} - GC^{-1} \\ RL^{-1} - GC^{-1} & RL^{-1} + GC^{-1} \end{pmatrix}.$$

Here, $\xi = (\xi^+, \xi^-)$ with $\xi^\pm(\cdot,\cdot) \in \mathbb{R}$ are the so-called characteristic variables, where ξ^+ represent right-traveling components and ξ^- represents left-traveling components. Inductance, capacitance, resistance and the admittance per unit length of the conductor are represented by the constant parameters L, C, R and G. The voltage and current can be computed from the characteristic variables [25].

2.1 The optimization problem

We would like to start by revisiting the optimization problem (5) of [27]. Therefore we introduce a network $\mathcal{G} = (V, A)$ with power sources $V_Q \subset V$ and customers $V_S \subset V$. Let δ_v^+ denote the set of outgoing arcs of vertex v and δ_v^- denote the set of ingoing arcs of vertex v, then $V_S = \{v \in V | |\delta_v^+| = 0\}$ and $V_Q = \{v \in V | |\delta_v^-| = 0\}$. Our aim is to minimize the mismatch between the load and the demand at all customers of the set V_S in the network by choosing the optimal inflow controls $u_q(t)$ for all power sources $q \in V_Q$. Furthermore the set δ_v is the set of all arcs adjacent to vertex v. We denote by $n = |V|$ the number of vertices of graph $\mathcal{G}$ and by $m = |A|$ the number of arcs. We interpret each arc r as an interval $I_r = [0, l_r]$ with length l_r and define the distribution at inner vertices through constant distribution matrices $\boldsymbol{D}^\pm \in \mathbb{R}^{m \times m}$. We assemble the $\xi(\cdot,\cdot)$ of all arcs to a vector $\boldsymbol{\xi}$ and all power inflows $u_q(t)$ to a vector $\boldsymbol{u}$. Furthermore matrices $\boldsymbol{\Lambda}$ and $\boldsymbol{B}$ are composed of the matrices Λ and B of the different arcs in order to formulate the transport along the arcs and the coupling and boundary conditions in compact form

$$\partial_t \boldsymbol{\xi} + \boldsymbol{\Lambda} \partial_x \boldsymbol{\xi} + \mathbf{B} \boldsymbol{\xi} = 0 \tag{2}$$

and

$$\begin{pmatrix} \boldsymbol{\Lambda}^+ & 0 \\ 0 & \mathbf{D}^- \end{pmatrix} \boldsymbol{\xi}(0,t) = \begin{pmatrix} \mathbf{D}^+ & 0 \\ 0 & \boldsymbol{\Lambda}^- \end{pmatrix} \boldsymbol{\xi}(l,t) + \begin{pmatrix} \boldsymbol{\Lambda}^+ & 0 \\ 0 & 0 \end{pmatrix} \boldsymbol{u}(t), \tag{3}$$

where $\Lambda^\pm$ handle the possibly different properties of the transmission lines.

Furthermore we assume fixed values as initial conditions

$$\xi_r(x,0) = \xi_0(x) \in \mathbb{R} \qquad \forall r \in A, x \in [0, l_r]. \tag{4}$$

In order to measure the mismatch between the transmitted load and the demand $Q_s(t), s \in V_S$ for a time interval $[0,T]$, we define the load at customer s at time t as

$$C_s(t) = \sum_{i \in \delta_s} \xi_i^+(l_i, t).$$

Finally, we can formulate the optimization problem (5).

$$\begin{cases} \min\limits_{\boldsymbol{u} \in \mathcal{U}} F(\boldsymbol{\xi}, \boldsymbol{u}) = \frac{1}{2} \sum\limits_{s \in V_S} \int_0^T \left(Q_s(t) - C_s(t)\right)^2 dt \\ \\ \text{s.t. } (2), (3) \text{ and } (4), \end{cases} \tag{5}$$

where $\boldsymbol{u}$ can be chosen from the set of feasible inflow controls $\mathcal{U}$. For more detailed derivations of the optimization problem, we refer to [25, 27].

The theoretical results on the existence of optimal controls for this type of problems can be found in [15]. Therein the authors prove the existence of an optimal control for a system of nonlinear conservation laws with a source term under certain conditions. These results apply to problem (5).

Next, we introduce switches in the form of binary variables $sw_r(t)$ for the telegraph equations and extend the power network flow problem (5). The ideas are universally applicable and can be applied to other network frameworks with time-dependent variables as well. Without binary decisions the distribution at a vertex is predetermined at all points in time, cf. equation (3). We influence the distribution at crossings by disabling certain arcs or subgrids. This means that if arc $r = (v_1, v_2)$ were to be disabled at time t, we would need to ensure that no flow from arcs entering nodes v_1 or v_2 is distributed to $\xi_r^+(0,t)$ or $\xi_r^-(l_r,t)$, respectively.

In Figure 1 we see a network with two arcs (marked with a hatched circle) that can be temporarily disabled, namely arc 4 and arc 5. If arc 4 is disabled, then all load ξ^+ that reaches vertex 4 will be distributed to arcs 2 and 3 and all load ξ^- that reaches vertex 4 will be passed on to arc 1.

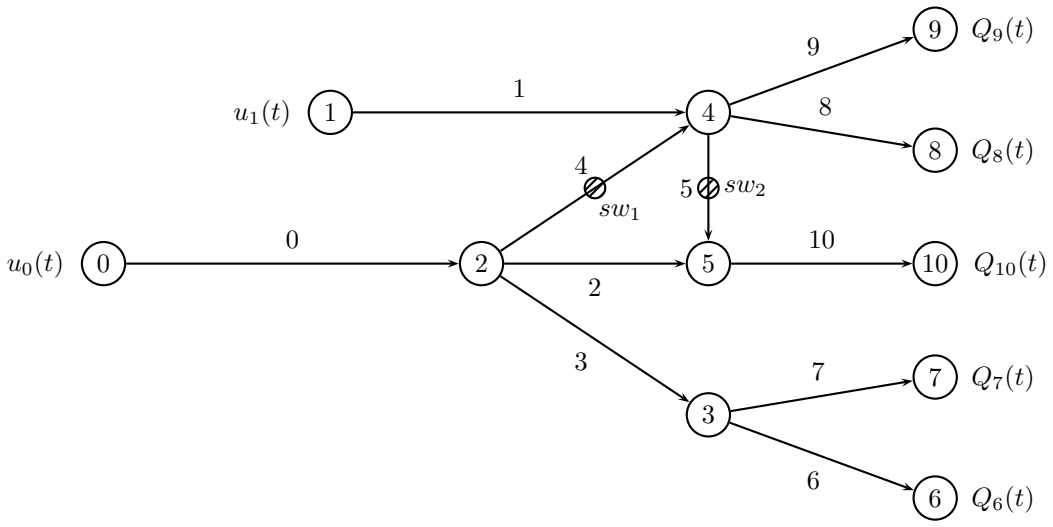

Figure 1: Extended tree-network with two switches (hatched circles).

In partial outer convexification [48, 50, 51] we do not consider single switches. Here, binary variables are defined that contain the information about all switches. We will call these binaries outer controls, to avoid confusion with the single switches. A finite set of configurations Ω of different outer controls is predetermined and the corresponding binary variables $b_c(t), c \in \Omega$ are introduced. Considering the graph of figure 1 there are four outer controls. The first control implies that arcs 4 and 5 are active, the second control demands inactivity of both arcs and controls three and four demand that exactly one of the arcs is active. In general any combination of the states of all switches is possible and therefore there are $|\Omega| = 2^{n_{sw}}$ outer controls, where n_{sw} denotes the number of switchable arcs. The number of outer controls can be reduced if constraints connecting different switches are known. For example if for some reason switch 1 has to be on if the switches 2 and 3 are off, then any outer control contradictory to this relation can be neglected. This is reasonable for example in the context of traffic light control [26], where accidents can be caused by a wrong combination of green phases of different traffic lights, but also in the case of power networks where it can be important to prevent overvoltage for example. We will denote the number of outer controls as $n_{oc} = |\Omega|$. For each outer control c constant distribution matrices $\mathbf{D}_c^+ = ({}^c d_{rk}^+) \in \mathbb{R}^{m\times m}$ and $\mathbf{D}_c^- = ({}^c d_{rk}^-) \in \mathbb{R}^{m\times m}$ are defined. When applying partial outer convexification we substitute (3) by

$$\begin{pmatrix} \mathbf{\Lambda}^+ & 0 \\ 0 & \sum\limits_{c\in\Omega} b_c(t)\mathbf{D}_c^- \end{pmatrix} \boldsymbol{\xi}(0,t) = \begin{pmatrix} \sum\limits_{c\in\Omega} b_c(t)\mathbf{D}_c^+ & 0 \\ 0 & \mathbf{\Lambda}^- \end{pmatrix} \boldsymbol{\xi}(l,t) + \begin{pmatrix} \mathbf{\Lambda}^+ & 0 \\ 0 & 0 \end{pmatrix} \boldsymbol{u}(t). \tag{6}$$

Since an outer control c includes information about all switches at a given time t, we know that exactly one control can be active simultaneously, which results in so-called SOS constraints of type 1:

$$\sum_{c\in\Omega} b_c(t) = 1 \qquad \forall t \in [0,T]. \tag{7}$$

The optimization problem becomes

$$\begin{cases} \min\limits_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{b}\in\mathcal{B}} F(\boldsymbol{\xi},\boldsymbol{u},\boldsymbol{b}) = \frac{1}{2} \sum\limits_{s\in V_S} \int_0^T \left(Q_s(t) - C_s(t)\right)^2 dt \\ \\ \text{s.t. } (2), (4), (6) \text{ and } (7), \end{cases} \tag{8}$$

where $\boldsymbol{b} \in \mathcal{B}$ if $b_c(t) \in \{0,1\}$ for all $t \in [0,T]$ and all $c \in \Omega$.

So far we are able to deactivate and activate single arcs. In applications it can be of interest to deactivate whole subgrids simultaneously. Therefore we need to couple the switching decision for a set of arcs. Let us first formally define a subgrid.

Definition 2.1. *Let $\alpha(r)$ denote the start vertex of arc r and $\omega(r)$ denote the end vertex of arc r. A subgrid $\mathcal{G}_{Sub} = (V_{Sub}, A_{Sub})$ of the graph $\mathcal{G} = (V, A)$ fulfills the following conditions*

1. $V_{Sub} \subset V$
2. $A_{Sub} \subset A$
3. $r \in A : \alpha(r), \omega(r) \in V_{Sub} \Longleftrightarrow r \in A_{Sub}$
4. $v \in V : (\alpha(r) \in V_{Sub} \ \forall\ r \in A : \omega(r) = v)$
   and $(\omega(r) \in V_{Sub} \ \forall\ r \in A : \alpha(r) = v) \Longrightarrow v \in V_{Sub}$
5. *G_{Sub} is connected, i.e. $\forall$ pairs of vertices in V_{Sub} there exists a connecting path in the underlying undirected graph.*

The first two conditions simply imply that all vertices and arcs of the subgrid have to be contained in the original graph. The third condition demands that if there is an arc $r \in A$ such that both endpoints are contained in V_{Sub} then r is also contained in the arc set A_{Sub} of the

subgrid. And that the endpoints of an arc of the arc set A_{Sub} of the subgrid have to be part of the subgrid, too. The fourth condition demands that if a vertex is only connected through arcs whose other endpoints lie in the vertex set V_{Sub} of the subgrid, then this vertex has to be part of the subgrid.

An example of a subgrid is given in Figure 2. We intend to activate and deactivate the subgrid G_{Sub} consisting of vertices $4, 5, 6$ and all arcs inside of the gray area. Therefore we have to change the state of the switches sw_1, that correspond to the arcs entering and leaving the subgrid G_{Sub}.

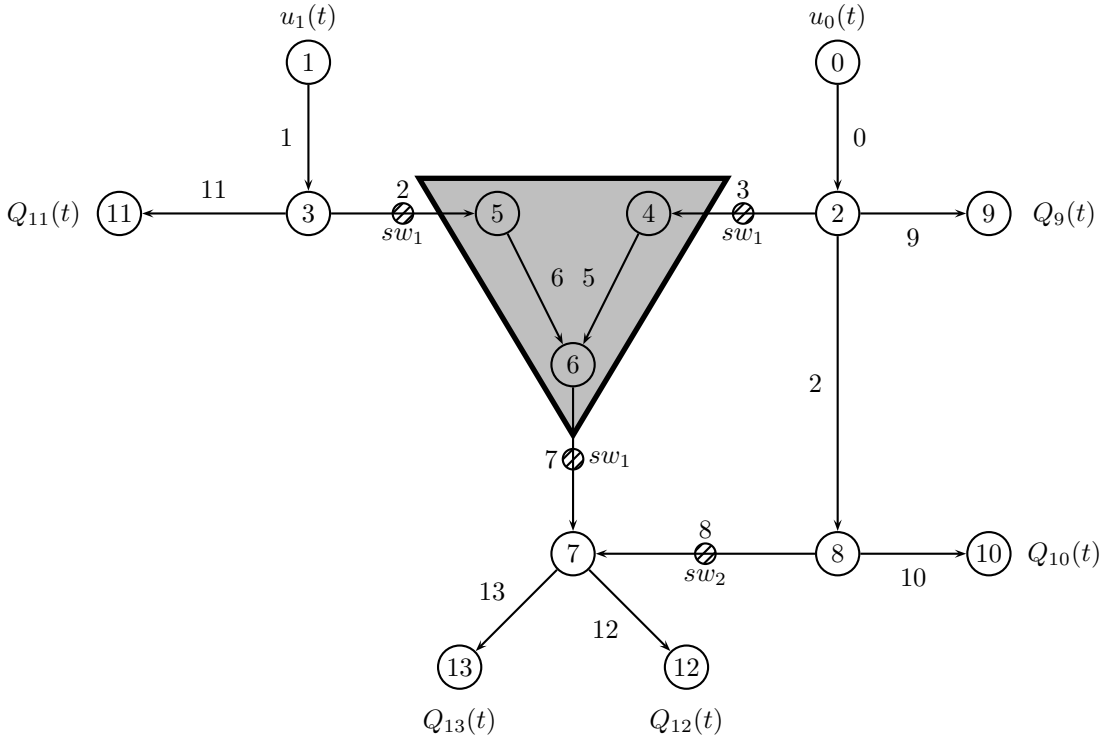

Figure 2: Network with subgrid in gray scale and two switches.

Problem (8) is a nonlinear optimization problem and the bilinear equality constraints lead to a nonconvex feasible set. We will apply a first discretize then optimize strategy, where these constraints can easily be substituted by linear constraints in the discretized problem. The discretized version of problem (8) is a MINLP. We will later show how to benefit from properties of partial outer convexification in a solution method. But first we will discretize (8) and introduce additional constraints to obtain more realistic controls.

2.2 The discretized optimization problem

In the following we will derive an appropriate discretized version of (8) as a basis for the optimization of power networks.

First of all, we introduce step sizes Δt and Δx in time and space for the discretization of equation (2) which satisfy the CFL-condition

$$\Delta t \leq \sqrt{LC}\Delta x. \tag{9}$$

The continuous time interval $[0, T]$ becomes $\{t_0, \ldots, t_{n_T}\}$ where $n_T = \frac{T}{\Delta t}$ is the number of discrete time steps and $t_e = e\Delta t, e = 0, \ldots, n_T$. On each arc we get the discretization points $\{x_0, \ldots, x_{\tilde{l}_r}\}$

with $x_d = d\Delta x, d = 0, \ldots, \tilde{l}_r$. The quotient of the arc lengths and the space step size $\tilde{l}_r = \frac{l_r}{\Delta x}$ is the number of discretization points of arc r. We assume that n_T and $\tilde{l}_r$ are integer for all $r \in A$.

Then, we exploit that the system (1) is written in its characteristic variables. Therefore information from only one side is relevant and a simple upwind scheme can be applied as in [27]. This yields

$$\begin{aligned} \xi^+(x, t+\Delta t) = \quad & \xi^+(x,t) - \lambda^+ \tfrac{\Delta t}{\Delta x}(\xi^+(x,t) - \xi^+(x-\Delta x, t)) \\ & -b_{11}\xi^+(x,t)\Delta t - b_{12}\xi^-(x,t)\Delta t \end{aligned} \tag{10}$$

and

$$\begin{aligned} \xi^-(x, t+\Delta t) = \quad & \xi^-(x,t) - \lambda^- \tfrac{\Delta t}{\Delta x}(\xi^-(x+\Delta x, t) - \xi^-(x,t)) \\ & -b_{12}\xi^+(x,t)\Delta t - b_{11}\xi^-(x,t)\Delta t, \end{aligned} \tag{11}$$

the discretized form of the system (1) for the transport along a single arc. Additionally, we apply a quadrature formula to the objective function of (8), which results in

$$F(\boldsymbol{\xi}(\boldsymbol{u}), \boldsymbol{u}) = \frac{1}{2} \sum_{s \in V_S} \sum_{e=0}^{n_T} w_e \left(Q_s(t_e) - C_s(t_e)\right)^2. \tag{12}$$

A trapezoidal rule for which the weights are $w_e = \Delta t$ and $w_0 = \frac{1}{2}\Delta t$, $w_T = \frac{1}{2}\Delta t$, respectively, is a natural choice. Nevertheless, for our computations we will simply set $w_e = \Delta t$ for all $e = 0, \ldots, n_T$.

Furthermore, we substitute the bilinear equality constraints by a set of linear constraints. The MINLP has become a mixed-integer quadratic problem (MIQP).

2.3 Additional Constraints that couple over time

In practice additional constraints are often necessary to achieve applicable results. Thus they have to be added to problem (8). As mentioned above, we will solve these problems in a first discretize then optimize manner. Therefore we will introduce the additional constraints for the discretized version.

Without further constraints it is not unlikely that a switch changes its state rapidly between 0 and 1. We would like to prevent such behavior and give a minimum time between two alterations. There are different ideas for the implementation, which are commonly used in the modeling of traffic lights [26, 28]. We can choose a coarser time grid for the binary variables than for the inflow controls and state variables. Furthermore, we can introduce a penalty term in the objective function to penalize altering the state. Nevertheless, we will introduce additional constraints for our implementation.

To this end, we introduce vectors $a^c \in \{0,1\}^{n_{sw}}$ with $c \in \Omega$, which contain one entry per switch. The j-th entry has value 0 if switch j is deactivated by the outer control c and 1 if it is activated. There are two types of constraints. The first inequality

$$\sum_{e=k+1}^{k+\left\lfloor \frac{M_1}{\Delta t} \right\rfloor} \sum_{c \in \Omega} a_j^c b_c(t_e) \geq \sum_{c \in \Omega} a_j^c \left\lfloor \frac{M_1}{\Delta t} \right\rfloor \left(-b_c(t_k) + b_c(t_{k+1})\right) \tag{13}$$

for all $k \leq n_T - \left\lfloor \frac{M_1}{\Delta t} \right\rfloor, j \in \{1, \ldots, n_{sw}\}$ ensures that switch j has to keep its state for at least a time of M_1 after switching from off to on. Whereas the second inequality

$$\sum_{e=k+1}^{k+\left\lfloor \frac{M_2}{\Delta t} \right\rfloor} \left(1 - \sum_{c \in \Omega} a_j^c b_c(t_e)\right) \geq \sum_{c \in \Omega} a_j^c \left\lfloor \frac{M_2}{\Delta t} \right\rfloor \left(b_c(t_k) - b_c(t_{k+1})\right) \tag{14}$$

for all $k \leq n_T - \left\lfloor \frac{M_2}{\Delta t} \right\rfloor, j \in \{1, \ldots, n_{sw}\}$ ensures that the switch has to keep its state for a time of at least M_2 after switching from on to off.

3 Solution techniques

Solving nonlinear optimization problems and mixed integer programming problems can both be a challenging task. In mixed-integer nonlinear programming [5, 31, 39] the difficulties of handling integer variables and dealing with nonlinear functions are combined. Even though there has been a lot of research so far, there is still need for improvements. Many publications consider branch and bound strategies for nonlinear convex problems [10, 13, 17, 29, 40, 53, 55]. Others are dealing with multitree methods for convex problems that aim to reduce the number of NLPs that have to be solved. Common multitree methods are outer approximation [11, 19, 21] and the generalized Benders decomposition [24, 56].

In general nonconvex problems are harder to solve. One common approach is to approximate the nonlinear functions by piecewise linear functions such that an MILP solver can be applied [23, 57, 58]. Many other publications focus on branch and bound for nonconvex MINLP, so-called spatial branch and bound. Here the focus lies on different relaxation techniques [1, 6, 42]. Finally, there has also been a lot of research concerning heuristics [8, 12, 43].

For an efficient algorithm it is important to benefit from the structure of an MINLP if possible.

3.1 Two-step heuristic

In the following we describe how partial outer convexification can help during the solution process. Partial outer convexification first appeared in the context of control problems constrained by ordinary differential equations [48]. In various publications [35, 36, 49, 50] a two-step heuristic has been derived for ODE-constrained problems, which divides the solution of the MINLP in an NLP and an MILP without dynamic constraints. Furthermore, an approximation theorem has been derived, giving an upper bound on the error of the heuristic [52]. A first publication on the topic of problems constrained by semilinear PDEs with a bound on the error has been achieved in [33] with the help of semigroups. Unfortunately these results demand an amount of smoothness which usually only occurs in the case of parabolic PDEs. In [32] approximation results have been published for first order semilinear hyperbolic PDEs in one space dimension. Recently, the two-step heuristic has been successfully applied to traffic light optimization in road networks [26] described by the well-known Lighthill–Williams–Richards model containing hyperbolic PDEs. Here, a discretized version of the approximation theorem has been derived to which we will refer later.

In the following, we will lay the theoretical foundation for the application of the two-step heuristic to problem (8). The general idea is to approximate the solution of the MINLP by first solving its NLP-relaxation and afterwards applying a rounding strategy to fulfill the integrality condition. In a final step, the corresponding trajectories to the rounded binary values can be computed by a simple simulation. The procedure has been summarized in algorithm 1.

Algorithm 1 Two-step heuristic

1: Discretize problem (8) with appropriate step sizes Δt and Δx fulfilling the CFL-condition (9).

2: **Step 1**: Relax the integrality conditions

$$b_c(t_e) \in \{0,1\} \to \tilde{b}_c(t_e) \in [0,1] \ \forall e \in \{0,\dots,n_T\}, c \in \Omega$$

and solve the arising NLP. This yields controls $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{b}}$.

3: **Step 2**: Compute a feasible binary solution $\boldsymbol{b}$ out of $\tilde{\boldsymbol{b}}$ by a suitable rounding procedure

4: Get the feasible trajectories $\boldsymbol{\xi}^{\pm}(t)$ and the objective function value $F(\boldsymbol{\xi}, \boldsymbol{u}, \boldsymbol{b})$ by a simulation with controls $\tilde{\boldsymbol{u}}$ and $\boldsymbol{b}$.

This approach is based on a theorem which yields an upper bound on the error between the trajectories of the relaxed problem and those of the solution obtained by the heuristic. We state

a discretized version of the original theorem taken from [26]. Let

$$\mathcal{H} = \left\{ \hat{\alpha} \in \mathbb{R}^{n_{oc}}_{\geq 0} \,\middle|\, \sum_{i=1}^{n_{oc}} \hat{\alpha}_i = 1 \right\}$$

be the set of all relaxed controls that fulfill the SOS type 1 conditions (7). Furthermore we need two norms $\|\cdot\|_X : \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ and $\|\cdot\|_\Omega : \mathbb{R}^{n_{oc}} \to \mathbb{R}_{\geq 0}$.

Theorem 3.1. [26] *Let $\mathcal{I} = \{0, \dots, n_T - 1\}, \mathcal{D} \subseteq \mathbb{R}^N$, and $\Phi : \mathcal{I} \times \mathcal{D} \to \mathbb{R}^{n \times n_{oc}}$ be a matrix-valued function that is continuous with respect to the second argument and satisfies*

$$\|\Phi(t,\mu)\nu\|_X \leq M_{oc}\|\nu\|_\Omega \qquad \forall t \in \mathcal{I}, \mu \in \mathcal{D}, \nu \in \mathbb{R}^{n_\Omega}, \tag{15}$$

$$\|(\Phi(t,\mu) - \Phi(t,\eta))\hat{\alpha}\|_X \leq L_{oc}\|\mu - \eta\|_X \qquad \forall t \in \mathcal{I}, \mu, \eta \in \mathcal{D}, \hat{\alpha} \in \mathcal{H} \tag{16}$$

for constants $M_{oc}, L_{oc} < \infty$. Furthermore, for each $t \in \mathcal{I}$ let $h_t > 0, \hat{\alpha}^t, \beta^t \in \mathcal{H}$ such that $T = \sum_{t=0}^{n_T-1} h_t$ and for each $t \in \mathcal{I} \cup \{n_T\}$ let $\mu^t, \eta^t \in \mathcal{D}$ be given such that for all $t \in \mathcal{I}$

$$\mu^{t+1} = \mu^t + h_t \Phi(t, \mu^t)\hat{\alpha}^t \qquad \text{and} \qquad \eta^{t+1} = \eta^t + h_t \Phi(t, \eta^t)\beta^t.$$

If for some set $\mathcal{I}' \subseteq \mathcal{I}$, constants $C_{oc}, \epsilon < \infty$ and some vector $\delta^0 \in \mathbb{R}^{n_{oc}}$ it holds that

$$\|(\Phi(t+1, \mu^{t+1}) - \Phi(t, \mu^t))\nu\|_X \leq h_t C_{oc}\|\nu\|_\Omega \qquad \forall t \in \mathcal{I}', \nu \in \mathbb{R}^{n_{oc}}, \tag{17}$$

$$\left\| \delta^0 + \sum_{t=0}^{k-1} h_t(\hat{\alpha}^t - \beta^t) \right\|_\Omega \leq \epsilon \qquad \forall k \in \mathcal{I} \cup \{n_T\}, \tag{18}$$

then it follows with $T'_k = k \max\{h_t | t = 0, \dots, k-1\}$ and $n_{jump} = |\mathcal{I} \setminus (\mathcal{I}' \cup \{n_T - 1\})|$ that for all $k \in \mathcal{I} \cup \{n_T\}$

$$\sum_{t=0}^{k} h_t \|\mu^t - \eta^t\|_X \leq \frac{\exp(T'_k L_{oc} - 1)}{L_{oc}} (\|\mu^0 - \eta^0\|_X + (2M_{oc}(1 + n_{jump}) + TC_{oc})\epsilon). \tag{19}$$

Theorem 3.1 is the basis of the two-step heuristic. It delivers an upper bound on the produced error and furthermore, we observe in equation (19) that the mismatch of the trajectories is forced to zero for $\epsilon \to 0$ if $\mu^0 = \eta^0$. Therefore, we are interested in the minimization of ϵ. This gives rise to a rounding strategy.

In mixed-integer optimization rounding is in general a bad idea. In many cases it leads to suboptimal or even infeasible solutions. In the context of partial outer convexification on the other hand it has been proven to be an efficient way to avoid the application of a time-consuming branch and bound strategy. Based on Theorem 3.1 the Combinatorial Integral Approximation Problem (CIAP) [35] has been derived. In Theorem 3.1 the quality of the approximation depends linearly on the value of ϵ in condition (18). Therefore we are interested in minimizing ϵ, which is the aim of the CIAP. We will state the version from [26].

Let $\hat{\omega}^t$ be the values of the binary variables in time step t of the solution of the NLP. The CIAP has the following form:

$$\begin{aligned} \min_{\substack{\beta^t \in \mathcal{H} \cap \{0,1\}^{n_{oc}} \\ t=0,\dots,n_T-1 \\ \delta^0 \in \mathbb{R}^{n_{oc}}, \epsilon \in \mathbb{R}}} \quad & \epsilon \\ \text{s.t.} \quad & \left\| \delta^0 + \sum_{t=0}^{k-1} \Delta t(\bar{\omega}^t - \beta^t) \right\|_\Omega \leq \epsilon, \quad \text{for all } k = 0, \dots, n_T. \end{aligned} \tag{20}$$

Remark: Additional constraints, e.g. (13-14), can be included in (20).

With the choice $\|\cdot\|_\Omega = \|\cdot\|_\infty$ the CIAP can be transformed into an MILP by applying standard linearization techniques. An upper bound on the value of ϵ in terms of Δt has been

derived in [50] and therefore an upper bound on the mismatch in equation (19) of Theorem 3.1. In order to apply the two-step heuristic to problem (8) we need to derive a vectorial shortform

$$\xi(\cdot, t_{e+1}) = \xi(\cdot, t_e) + \Delta t \Phi^c(t, \xi), \tag{21}$$

where the function Φ^c depends on the binary variables. Afterwards we assure that all requirements of theorem 3.1 are fulfilled.

We start by transforming (10) and (11) to

$$\begin{aligned} \xi_r^+(x_d, t_{e+1}) = \quad & \xi_r^+(x_d, t_e) + \Delta t[-\lambda_r^+ \tfrac{1}{\Delta x}(\xi_r^+(x_d, t_e) - \Theta_c^+(\boldsymbol{\xi}(\cdot, t_e), r, x_d, t_e)) \\ & -b_{11}\xi_r^+(x_d, t_e) - b_{12}\xi_r^-(x_d, t_e)] \end{aligned} \tag{22}$$

and

$$\begin{aligned} \xi_r^-(x_d, t_{e+1}) = \quad & \xi_r^-(x_d, t_e) + \Delta t[-\lambda_r^+ \tfrac{1}{\Delta x}(\xi_r^-(x_d, t_e) - \Theta_c^-(\boldsymbol{\xi}(\cdot, t_e), r, x_d, t_e)) \\ & -b_{11}\xi_r^-(x_d, t_e) - b_{12}\xi_r^+(x_d, t_e)]. \end{aligned} \tag{23}$$

The functions Θ_c^+ and Θ_c^- are defined as

$$\Theta_c^+(\boldsymbol{\xi}(\cdot, t_e), r, x_d, t_e) = \begin{cases} \sum_{k \in \delta^-_{\alpha(r)}} {}^c d^+_{rk} \frac{\lambda_k^+}{\lambda_r^+} \xi_k^+(x_{\tilde{l}_k}, t_e) & \text{if } d = 1, r \in A \setminus A_Q \\ u_r(t_e) & \text{if } d = 1, r \in A_Q \\ \xi_r^+(x_{d-1}, t_e) & \text{otherwise} \end{cases} \tag{24}$$

and

$$\Theta_c^-(\boldsymbol{\xi}(\cdot, t_e), r, x_d, t_e) = \begin{cases} \sum_{k \in \delta^+_{(}\omega(r))} {}^c d^-_{rk} \frac{\lambda_k^-}{\lambda_r^-} \xi_k^-(x_{\tilde{l}_k}, t_e) & \text{if } d = \tilde{l}_r - 1, i \in A \setminus A_S \\ 0 & \text{if } d = \tilde{l}_r - 1, r \in A_S \\ \xi_i^-(x_{d+1}, t_e) & \text{otherwise} \end{cases}. \tag{25}$$

Equations (22 - 25) define the vectorial shortform (21). Next, we will prove an important stability property of the network flow problem, which will play an important part when showing that we can fulfill all necessary inequalities for the application of Theorem 3.1. In this context we will need the following norm.

$$||F||_{\infty,n} = \sum_{i=1}^{n} \sup_k |F_i(x_k)| \quad \text{with } x_k = k\Delta x,$$

where $F : \mathbb{R}^N \to \mathbb{R}^n$.

We can apply it to $(\xi_r^+(\cdot, t), \xi_r^-(\cdot, t))^T$. On a single edge this reads

$$\left\| \begin{pmatrix} \xi_r^+(\cdot, t_e) \\ \xi_r^-(\cdot, t_e) \end{pmatrix} \right\|_{\infty,2} = \sup_j |\xi_r^+(x_j, t_e)| + \sup_k |\xi_r^-(x_k, t_e)|.$$

Next, we want to show that our numerical scheme (10) and (11) is stable in the $\|\cdot\|_{\infty,2}$-norm under some conditions, i.e.

$$\left\| \begin{pmatrix} \xi_r^+(\cdot, t_{e+1}) \\ \xi_r^-(\cdot, t_{e+1}) \end{pmatrix} \right\|_{\infty,2} \leq \left\| \begin{pmatrix} \xi_r^+(\cdot, t_e) \\ \xi_r^-(\cdot, t_e) \end{pmatrix} \right\|_{\infty,2}$$

Lemma 3.2. *The numerical scheme (10) and (11) is stable in the $\|\cdot\|_{\infty,2}$-norm if*

$$\Delta t \leq \frac{1}{\lambda_r^+ + \Delta x b_{11}} \Delta x$$

is fulfilled.

Proof. Let j, k be arbitrary but fixed. Then, the numerical discretization leads to the estimate

$$
\begin{aligned}
&|\xi_r^+(x_j, t_{e+1})| + |\xi_r^-(x_k, t_{e+1})| \\
&= |\xi_r^+(x_j, t_e)(1 - \lambda_r^+ \frac{\Delta t}{\Delta x} - b_{11}\Delta t) + \lambda_r^+ \frac{\Delta t}{\Delta x}\xi_r^+(x_{j-1}, t_e) - b_{12}\Delta t \xi_r^-(x_j, t_e)| \\
&\quad + |\xi_r^-(x_k, t_e)(1 - \lambda_r^+ \frac{\Delta t}{\Delta x} - b_{11}\Delta t) + \lambda_r^+ \frac{\Delta t}{\Delta x}\xi_r^-(x_{k+1}, t_e) - b_{12}\Delta t \xi_r^+(x_k, t_e)| \\
&\overset{(*)}{\leq} \sup_j |\xi_r^+(x_j, t_e)|(1 - \lambda_r^+ \frac{\Delta t}{\Delta x} - b_{11}\Delta t) + \lambda_r^+ \frac{\Delta t}{\Delta x} \sup_j |\xi_r^+(x_{j-1}, t_e)| + b_{12}\Delta t \sup_k |\xi_r^-(x_k, t_e)| \\
&\quad + \sup_k |\xi_r^-(x_k, t_e)|(1 - \lambda_r^+ \frac{\Delta t}{\Delta x} - b_{11}\Delta t) + \lambda_r^+ \frac{\Delta t}{\Delta x} \sup_k |\xi_r^-(x_{k+1}, t_e)| + b_{12}\Delta t \sup_j |\xi_r^+(x_j, t_e)| \\
&= (\sup_j |\xi_r^+(x_j, t_e)| + \sup_k |\xi_r^-(x_k, t_e)|)(1 - b_{11}\Delta t + b_{12}\Delta t) \\
&= (1 - b_{11}\Delta t + b_{12}\Delta t) \left\| \begin{pmatrix} \xi_r^+(x, t_e) \\ \xi_r^-(x, t_e) \end{pmatrix} \right\|_{\infty,2},
\end{aligned}
$$

where $(*)$ is valid if

$$
1 - \lambda_r^+ \frac{\Delta t}{\Delta x} - b_{11}\Delta t \geq 0 \Leftrightarrow \Delta t \left(\frac{\lambda_r^+}{\Delta x} + b_{11} \right) \leq 1 \Leftrightarrow \Delta t \leq \frac{1}{\lambda_r^+ + \Delta x b_{11}} \Delta x,
$$

which is a stronger assumption than the CFL-condition (9). To achieve stability we finally need to show that

$$
(1 - b_{11}\Delta t + b_{12}\Delta t) \leq 1.
$$

This is always fulfilled since

$$
0 \leq \frac{G}{C} \quad \Longleftrightarrow \quad \frac{1}{2}\left(\frac{R}{L} - \frac{G}{C} \right) \leq \frac{1}{2}\left(\frac{R}{L} + \frac{G}{C} \right) \quad \Longleftrightarrow \quad b_{12} \leq b_{11}.
$$

□

Assuming that all input controls $\mathbf{u}(t)$ are known, Lemma 3.2 ensures that there exists an upper bound $M \in \mathbb{R}$ such that

$$
|\xi_r^+(x_j, t_e)| + |\xi_r^-(x_k, t_e)| \leq M
$$

holds for all $r \in A, j, k \in \{0, \ldots, \tilde{l}_r\}$ and $e \in \{0, \ldots, n_T\}$. This bound depends on the inflow $\boldsymbol{u}$, the initial values $\xi_0^{\pm}$ and the shape of the network $\mathcal{G} = (V, A)$.

Next, we will consider the requirements of Theorem 3.1 to show its applicability to problem (8), similar to the approach in [26]. As norms we choose the discrete L^1-norm

$$
\|\xi(\cdot, t_e)\|_X = \sum_{r \in A} \sum_{d=1}^{\tilde{l}_r} |\xi_r^+(x_d, t_e)| \Delta x + \sum_{r \in A} \sum_{d=0}^{\tilde{l}_r - 1} |\xi_r^-(x_d, t_e)| \Delta x
$$

and the maximum norm $\|\cdot\|_\Omega = \|\cdot\|_\infty$.

We start with (15) and choose $e \in \{0, \dots, n_T\}, \mu = \xi$ and $\nu \in \mathbb{R}^{n_\Omega}$.

$$\begin{aligned}
\|\Phi(t_e,\xi)v\|_X &= \sum_{r\in A}\sum_{d=1}^{\tilde{l}_r}\Big|\sum_{c\in\Omega}\Big(\frac{\lambda_r^+}{\Delta x}(\Theta_c^+(\xi_r(\cdot,t_e),x_d,t_e)-\xi_r^+(x_d,t_e))\\
&\quad - b_{11}\xi_r^+(x_d,t_e)-b_{12}\xi_r^-(x_d,t_e)\Big)\nu_c\Big|\Delta x\\
&\quad + \sum_{r\in A}\sum_{d=0}^{\tilde{l}_r-1}\Big|\sum_{c\in\Omega}\Big(\frac{\lambda_r^+}{\Delta x}(\Theta_c^-(\xi_r(\cdot,t_e),x_d,t_e)-\xi_r^-(x_d,t_e))\\
&\quad - b_{11}\xi_r^-(x_d,t_e)-b_{12}\xi_r^+(x_d,t_e)\Big)\nu_c\Big|\Delta x\\
&\le \sum_{r\in A}\sum_{d=1}^{\tilde{l}_r}\Big(2\frac{\lambda_r^+}{\Delta x}M+b_{11}M+b_{12}M\Big)\sum_{c\in\Omega}|\nu_c|\Delta x\\
&\quad + \sum_{r\in A}\sum_{d=0}^{\tilde{l}_r-1}\Big(2\frac{\lambda_r^+}{\Delta x}M+b_{11}M+b_{12}M\Big)\sum_{c\in\Omega}|\nu_c|\Delta x\\
&= \Big(4\frac{\lambda_r^+}{\Delta x}+2b_{11}+2b_{12}\Big)Mn_{oc}\sum_{r\in A}(\tilde{l_r})\|\nu\|_\Omega =: M_{oc}\|\nu\|_\Omega
\end{aligned}$$

Next, we consider (16). Here, we set $\eta = (\eta^+, \eta^-)^T$ and $\hat{\alpha} \in \mathcal{H}$.

$$\begin{aligned}
&\|[\Phi(t_e,\xi)-\Phi(t_e,\eta)]\hat{\alpha}\|_X =\\
&= \sum_{r\in A}\sum_{d=1}^{\tilde{l}_r}\Big|\sum_{c\in\Omega}\Big(\frac{\lambda_r^+}{\Delta x}(\Theta_c^+(\xi_r(\cdot,t_e),x_d,t_e)-\Theta_c^+(\eta_r(\cdot,t_e),x_d,t_e)-\xi_r^+(x_d,t_e)+\eta_r^+(x_d,t_e))\\
&\qquad - b_{11}\xi_r^+(x_d,t_e)+b_{11}\eta_r^+(x_d,t_e)-b_{12}\xi_r^-(x_d,t_e)+b_{12}\eta_r^-(x_d,t_e)\Big)\hat{\alpha}_c\Big|\Delta x\\
&+ \sum_{r\in A}\sum_{d=0}^{\tilde{l}_r-1}\Big|\sum_{c\in\Omega}\Big(\frac{\lambda_r^+}{\Delta x}(\Theta_c^-(\xi_r(\cdot,t_e),x_d,t_e)-\Theta_c^-(\eta_r(\cdot,t_e),x_d,t_e)-\xi_r^-(x_d,t_e)+\eta_r^-(x_d,t_e))\\
&- b_{12}\xi_r^+(x_d,t_e)+b_{12}\eta_r^+(x_d,t_e)-b_{11}\xi_r^-(x_d,t_e)+b_{11}\eta_r^-(x_d,t_e)\Big)\hat{\alpha}_c\Big|\Delta x\\
&\le \Big(4\frac{\lambda_r^+}{\Delta x}+2b_{11}+2b_{12}\Big)n_{oc}\|\eta-\xi\|_X =: L_{oc}\|\eta-\xi\|_X
\end{aligned}$$

In a last step we address (17), where we additionally assume that there exist constants $C_\Theta \in \mathbb{R}$ and $n_{jump} \in \mathbb{N}$ such that for all $\Delta t > 0, r \in A$ and $d = 1$ or $d = \tilde{l} - 1$

$$|\Theta_c^\pm(\xi_r(\cdot,t_e),x_d,t_{e+1})-\Theta_c^\pm(\xi_r(\cdot,t_e),x_d,t_e)| \le \Delta t C_\Theta \tag{26}$$

holds for all $t \in \{0, \dots, n_T\}$ except for n_{jump} exceptions.

Exploiting (26), inequality (15) and the vectorial shortform (21), we derive the following in-

equality

$$
\begin{aligned}
&\|[\Phi(t_{e+1},\xi)-\Phi(t_e,\xi)]\nu\|_X\\
&=\sum_{r\in A}\sum_{d=1}^{\tilde{l}_r}\Big|\sum_{c\in\Omega}\Big(\frac{\lambda_r^+}{\Delta x}(\Theta_c^+(\xi_r(\cdot,t_e),x_d,t_{e+1})-\Theta_c^+(\xi_r(\cdot,t_e),x_d,t_e)-\xi_r^+(x_d,t_{e+1})+\xi_r^+(x_d,t_e))\\
&\qquad-b_{11}\xi_r^+(x_d,t_{e+1})+b_{11}\xi_r^+(x_d,t_e)-b_{12}\xi_r^-(x_d,t_{e+1})+b_{12}\xi_r^-(x_d,t_e)\Big)\nu_c\Big|\Delta x\\
&+\sum_{r\in A}\sum_{d=0}^{\tilde{l}_r-1}\Big|\sum_{c\in\Omega}\Big(\frac{\lambda_r^+}{\Delta x}(\Theta_c^-(\xi_r(\cdot,t_e),x_d,t_{e+1})-\Theta_c^-(\xi_r(\cdot,t_e),x_d,t_e)-\xi_r^-(x_d,t_{e+1})+\xi_r^-(x_d,t_e))\\
&\qquad-b_{12}\xi_r^+(x_d,t_{e+1})+b_{12}\xi_r^+(x_d,t_e)-b_{11}\xi_r^-(x_d,t_{e+1})+b_{11}\xi_r^-(x_d,t_e)\Big)\nu_c\Big|\Delta x\\
&\le\sum_{r\in A}\sum_{d=1}^{\tilde{l}_r}\Big|\sum_{c\in\Omega}\Big(\frac{\lambda_r^+}{\Delta x}(\Theta_c^+(\xi_i(\cdot,t_e),x_d,t_{e+1})-\Theta_c^+(\xi_r(\cdot,t_e),x_d,t_e)-\Delta t\ {}^{+}\Phi_d^c(t_e,\xi))\\
&\qquad-b_{11}\Delta t\ {}^{+}\Phi_d^c(t_e,\xi)-b_{12}\Delta t\ {}^{-}\Phi_d^c(t_e,\xi)\Big)\nu_c\Big|\Delta x\\
&+\sum_{r\in A}\sum_{d=0}^{\tilde{l}_r-1}\Big|\sum_{c\in\Omega}\Big(\frac{\lambda_r^+}{\Delta x}(\Theta_c^-(\xi_r(\cdot,t_e),x_d,t_{e+1})-\Theta_c^-(\xi_r,x_d,t_e)-\Delta t\ {}^{-}\Phi_d^c(t_e,\xi))\\
&\qquad-b_{12}\Delta t\ {}^{+}\Phi_d^c(t_e,\xi)-b_{11}\Delta t\ {}^{-}\Phi_d^c(t_e,\xi)\Big)\nu_c\Big|\Delta x\\
&\le\Delta t2\Big(\Big(2\frac{\lambda_r^+}{\Delta x}+b_{11}+b_{12}\Big)M_{oc}+\frac{\lambda_r^+}{\Delta x}C_\Theta n_\Omega\Big)\|\nu\|_\Omega=:\Delta tC_{oc}\|\nu\|_\Omega
\end{aligned}
$$

for all $e\in 0,\dots,n_T-1$ except for n_{jump} exceptions.

This concludes the proof, that Theorem 3.1 is applicable to problem (8). In [50] it has been proven that there is an upper bound on ϵ which depends on Δt. Therefore, we can drive the error to zero by decreasing Δt if we keep Δx constant. Note, that this does not lead to a violation of the CFL-condition (9). In [26] the authors propose the inclusion of the additional constraints in the CIAP (20) from the second phase, the reconstruction phase. Note, that in this case no upper bound for ϵ is known. Therefore we have no guarantee that the error goes to zero if we decrease Δt.

3.2 Third step

In Algorithm 1 we first solve an NLP where the integrality conditions of the binary variables have been relaxed. Afterwards we get binary variables by a suitable rounding strategy, where we suggest solving the CIAP and we receive the correct values of the state variables afterwards via a simulation. Even though we know that the error between the solution of the first step and the rounded solution converges to 0 for $\Delta t\to 0$ whilst keeping Δx fixed, the approximation can be nonsatisfactory for $\Delta t\gg 0$. Note, that in contrast to the traffic light optimization in [26], we also have to find the optimal inflow controls. After obtaining the binary values denoted by $oc_c(t_e)$ through rounding, the inflow controls $\tilde{u}_q(t_e)$ from step 1 will most likely not be the optimal inflow controls corresponding to $oc_c(t_e)$. Therefore it is reasonable to exchange the simulation by a third optimization step and compute the optimal inflow controls for $b_c(t_e)$. The procedure has been summarized in Algorithm 2.

As explained above this will improve the quality of the solution for higher values of Δt compared to Algorithm 1. For a fixed spatial step size Δx and $\Delta t\to 0$ the improvement in the third step compared to a simple simulation vanishes.

Algorithm 2 Three-step heuristic

1: Discretize problem (8) with appropriate step sizes Δt and Δx fulfilling the CFL-condition (9).

2: **Step 1**: Relax the integrality conditions

$$b_c(t_e) \in \{0,1\} \rightarrow \tilde{b}_c(t_e) \in [0,1] \ \forall e \in \{0,\dots,n_T\}, c \in \Omega$$

and solve the arising NLP. This yields controls $\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{b}}$.

3: **Step 2**: Compute a feasible binary solution $\boldsymbol{b}$ out of $\tilde{\boldsymbol{b}}$ by a suitable rounding procedure

4: **Step 3**: Solve a NLP to get the optimal inflow controls $\boldsymbol{u}$ to the binaries $\boldsymbol{b}$.

4 Numerical results

All computations of this chapter have been performed on a PC equipped with 16GB Ram and an Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz. All NLPs have been solved with Ipopt 3.12.7, wherein the derivatives have been provided if an analytical formulation was available. Otherwise they have been computed with ADOL-C 2.6.3. All MILPs and MIQPs have been solved using IBM(R) ILOG(R) CPLEX(R) Interactive Optimizer 12.7.0.0.

In general, we will set the parameters of the telegraph equations to $L = C = 1$, $R = 0.001$, and $G = 0.002$ and all computations will be based on the graphs from Figures 1 (referred to as "tree 2sw") and 2 (referred to as "subgrid"). Here, we choose arc lengths $l_r = 1$ for all $r \in A$.

As demand settings we use the standard load profiles from the Stromnetz Berlin GmbH* as real world data. The data set distinguishes between four different types of customers: households, industry, farms and a group with constant demand. The profiles for one day (23rd of September, 2014) are presented in Figure 3. The data shows accumulated values given every 15 minutes. This yields data for 96 time points which we extend by 8 time steps of 0 demand in the beginning.

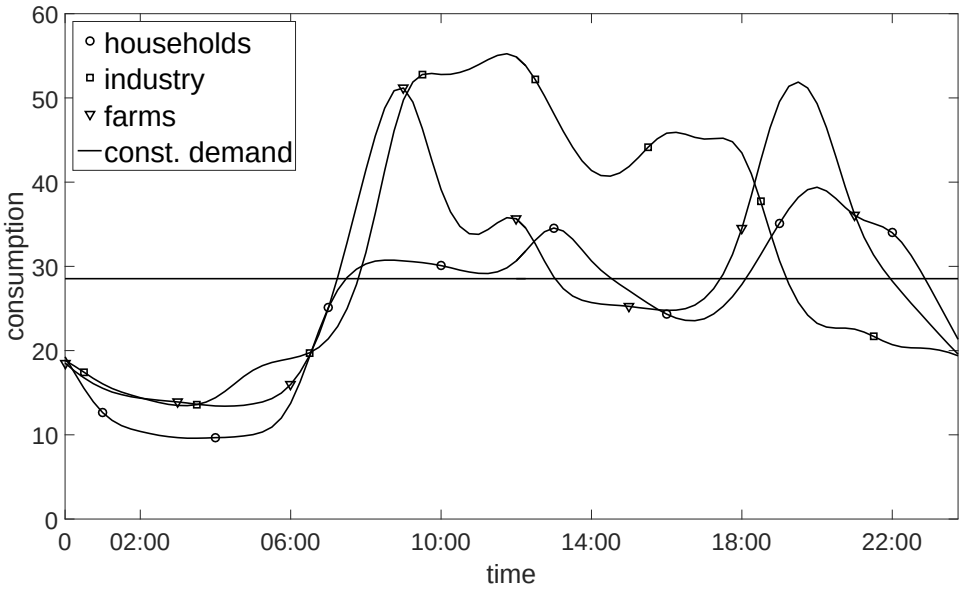

Figure 3: Standard load profiles for different types of customers.

Furthermore, we will set an upper bound on the inflow at every energy source. This will

*http://www.stromnetz-berlin.de/de/stromversorger.html

increase the dependence of the solution on the states of the switches.

4.1 Numerical examination of the three-step heuristic

In this section, we apply the two-step heuristic with and without an additional third step (Algorithm 1 and Algorithm 2, respectively) to various examples. As the rounding procedure we choose the CIAP (20). At first, we are interested in verifying the result from Theorem 3.1. Therefore, we choose attainable demands and decrease Δt whilst keeping Δx constant. Attainable demands are demands for which controls exist such that the demand is met with equality for all customers and all time steps. These demands have been computed through a simulation with inflow controls chosen as a constant value plus a sinusoidal function and binary controls obtained by rounding a sinusoidal function. The results are summarized in Table 1. OFV is short for objective function value and AC stands for "additional constraints". For the additional constraints (13) and (14), we demand that a switch has to keep its state for at least 4 time units. Note, that CPLEX has been stopped in step 2 after 1h if it was not able to terminate sooner.

Three-step heuristic

	Step 1		**Step 2**				**Step 3**	
Δt	OFV	time	AC	CIAP	OFV	time	OFV	time
Tree 2sw, $T = 15, \Delta x = 1$								
1	3.3e-4	0.79s	no	0.42	543.8	0.18s	46.9	0.06s
0.5	3.7e-3	2.88s	no	0.23	29.2	1.29s	3.29	0.26s
0.25	4.4e-3	13.51s	no	0.13	3.63	12.34s	0.20	1.47s
0.2	4.5e-3	23.18s	no	0.11	3.00	92.20s	0.24	2.46s
0.1	1.5e-3	179.6s	no	0.06	0.70	$>$ 1h	0.05	10.25s
1	3.3e-4	0.79s	yes	0.90	1080.1	0.35s	327.7	0.07s
0.5	3.7e-3	2.87s	yes	0.88	381.0	2.08s	44.8	0.31s
0.25	4.4e-3	13.5s	yes	0.87	515.0	59.4s	44.8	1.47s
0.2	4.5e-3	24.3s	yes	0.88	270.3	139.6s	34.2	2.93s
0.1	1.5e-3	188.9s	yes	0.84	282.0	460.0s	32.7	11.3s

Table 1: Effect of the discretization in the three-step heuristic.

In Figure 4, we visualize the results of Table 1 for the network of Figure 1. Note, that the objective function value after step 2 is computed via a simulation with the inflow controls from step 1 and the rounded binary controls from step 2. On the one hand, we see that we are able to drive the objective function value of the CIAP and thus ϵ to zero if no additional constraints are present. Furthermore, the objective function value of the controls obtained in step 2 goes to zero, too. On the other hand, this cannot be observed in the presence of additional constraints that couple over time. Here, the value of the CIAP only decreases very slowly and the overall objective function value is still not close to zero for $\Delta t = 0.1$. Furthermore, there is in both cases an immense benefit from the third step, even though its execution takes only a fraction of the time needed for step 1 and 2.

Note, that these results have to be handled with extra care. Choosing Δt independently from Δx leads to diffusion in the advection, even though the approximation scheme is still stable as long as the CFL-condition is fulfilled.

4.2 Comparison with a direct solver

Next, we consider the realistic demands from Figure 3. We assume that the first two consumers are households, that the third consumer is an industrial facility and that the fourth and fifth consumer represent farms. For Figure 2 only the order of the consumers has been altered. For the discretization we choose either a coarser grid with step sizes $\Delta x = \Delta t = 0.5$ or a finer grid with

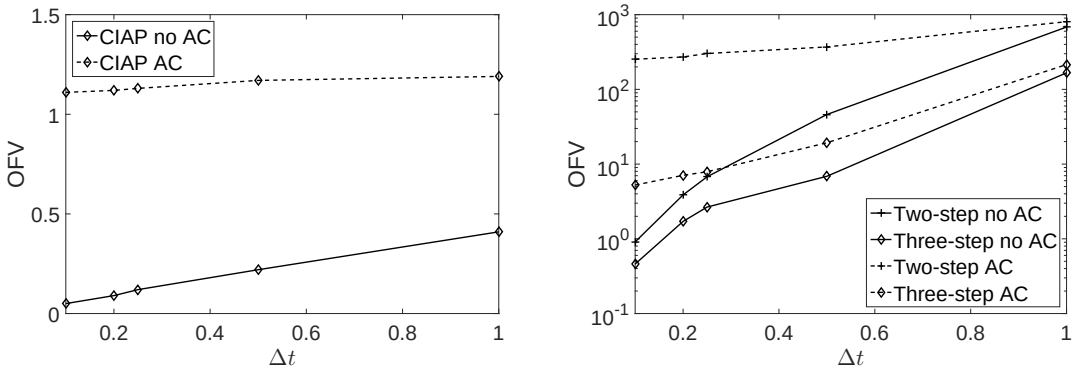

Figure 4: Visualization of the results of table 1 on network 1.

$\Delta x = \Delta t = 0.25$ with a time horizon of $T = 26$. Here, we compare the three-step heuristic to the performance of CPLEX when directly solving the MIQP that we derived previously.

The results of the three-step heuristic are summarized in Table 2. For every step, we saved the number of variables of the corresponding problem, the objective function value of its solution and the computing time. Note, that the number in the brackets denotes the number of binary variables. First of all, we observe, that the effect of the third step is still of high importance for realistic demands as it helps to decrease the objective function value considerably. Furthermore, it only requires a fraction of the computing time of the first step, which dominates the total computing time. The computing time of the CIAP immensely increases for bigger instances or finer discretization grids and especially for the computations with additional constraints. We further observe that for computations containing the subgrid, the presence of additional constraints can lead to a weak approximation after the second step, which cannot be balanced out by the third step. This becomes clear when we compare the objective function values of step 3 to the objective function values obtained by the MIQP solver (table 3).

Three-step heuristic

	Step 1		**Step 2**				**Step 3**		
#var	OFV	time	AC	#var	OFV	time	#var	OFV	time
Tree 2sw, $T = 26, \Delta x = 0.5, \Delta t = 0.5$									
306	2743	9.3s	no	209(204)	4633	0.41s	102	3071	1.0s
			yes	209(204)	4583	1.1s	102	3146	0.9s
Tree 2sw, $T = 26, \Delta x = 0.25, \Delta t = 0.25$									
606	2683	51.4s	no	409(404)	4688	5.5s	202	3057	6.7s
			yes	409(404)	4634	71.5s	202	3177	7.1s
Subgrid, $T = 26, \Delta x = 1, \Delta t = 0.5$									
306	1583	23.2s	no	209(204)	5811	0.7s	102	4760	1.3s
			yes	209(204)	12023	3.2s	102	8462	1.6s
Subgrid, $T = 26, \Delta x = 0.25, \Delta t = 0.25$									
606	1538	108.2s	no	409(404)	5638	54.9s	202	4712	13.5s
			yes	409(404)	12693	83.5s	202	8927	9.3s

Table 2: Computational results for the three-step heuristic.

When solving the MIQP with CPLEX we stop the computation after one hour and look at the objective function value and the gap. First of all we observe that CPLEX performed worse than the 3-step heuristic for almost every instance. It does not terminate within one hour and leaves a rather big gap. It especially performed badly for big instances such as the subgrid example on a fine grid.

Partial Outer Convexification MIQP

Δt	#var	AC	start	OFV	time	gap
Tree 2sw, $T = 26$						
0.5	5014(192)	no	cold	3739	> 1h	80.4%
0.5		no	warm	3071	> 1h	70.1%
0.5	5027(204)	yes	cold	3339	> 1h	54.4%
0.5		yes	warm	2939	> 1h	43.5%
0.25	14294(384)	no	cold	4316	> 1h	89.6%
0.25		no	warm	3057	> 1h	84.8%
0.25	14314(404)	yes	cold	4422	> 1h	71.0%
0.25		yes	warm	3177	> 1h	51.8%
Subgrid, $T = 26$						
0.5	6626(192)	no	cold	4585	> 1h	84.0%
0.5		no	warm	4760	> 1h	87.3%
0.5	6640(204)	yes	cold	7411	> 1h	87.7%
0.5		yes	warm	5246	> 1h	85.7%
0.25	18975(384)	no	cold	53432	> 1h	98.9%
0.25		no	warm	4214	> 1h	86.5%
0.25	18995(404)	yes	cold	53432^-	2140.8s	
0.25		yes	warm	8872^-	900.9s	

Table 3: Computational results for partial outer convexification.

Note that CPLEX has numerical difficulties in all computations that are marked with a minus in the OFV-column, which can not be solved by increasing the numerical emphasis and the Markowitz tolerance as recommended by the CPLEX user's manual. In these cases CPLEX declares a solution as optimal, which can be proven to be feasible, but not optimal. Moreover, CPLEX accepts a better solution as feasible in a warm start, which also contradicts the optimality of the nonoptimal solution and illustrates, that this behavior is an internal CPLEX issue. Generally CPLEX benefits from warm starts for bigger instances. However, in one of the cases the warm start led to a worse control. The warm starts are performed with the solutions obtained by using the three-step heuristic.

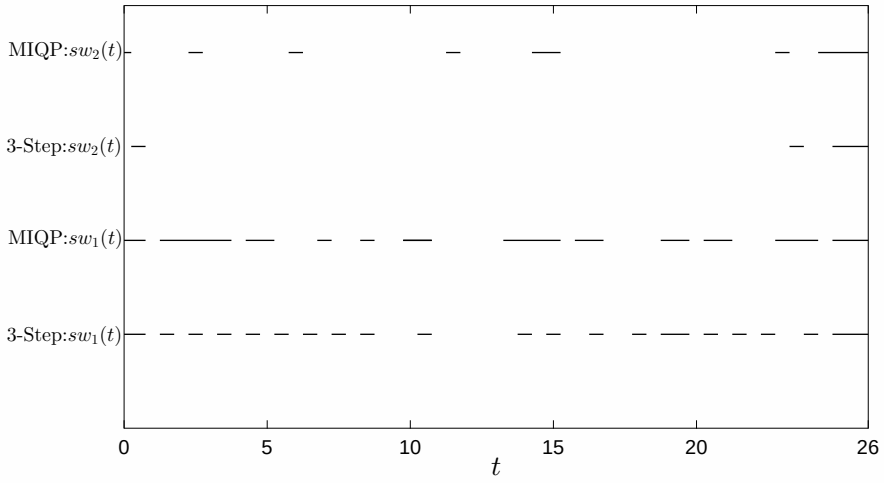

Figure 5: Switches obtained on network 1 with $\Delta t = \Delta x = 0.5$ without additional constraints. A black bar represents a phase in which the switch has value one.

Finally, we want to compare the controls produced by the three-step heuristic with the controls from the MIQP solver. Therefore we consider the computation with the extended tree network with a coarse grid and without additional constraints. Here, the three-step heuristic obtained an objective function value of 3071 and the MIQP solver an objective function value of 3739. The state of the switches is visualized in Figure 5 for both computations, whereas the inflow controls can be seen in Figure 6.

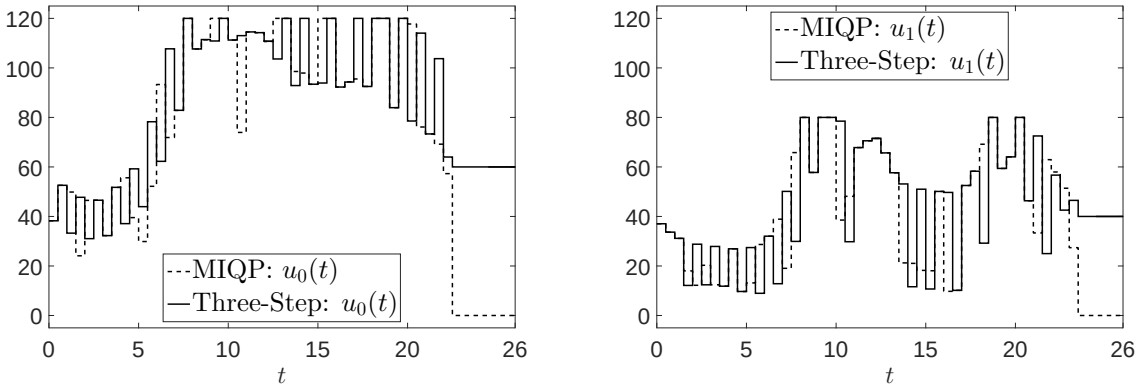

Figure 6: Inflow controls obtained on network 1 with $\Delta t = \Delta x = 0.5$ without additional constraints.

First of all we observe, that the solution computed with the three-step heuristic tends to alter the state of the switches more often. This behavior is then passed on to the inflow controls. Furthermore, we observe that if the binary controls of the two solutions coincide for some time steps, then the outflow controls coincide as well. Here, we see a time shift between the inflow controls, which is due to the difference in the lengths of the paths from the sources to the consumers.

5 Conclusion and future work

We have derived a model for the inflow control of power grids based on hyperbolic differential equations with the opportunity to temporarily disable single arcs or whole subgrids. In order to apply a two-step heuristic based on partial outer convexification, we proved the applicability of an important approximation result. Furthermore we extended the algorithm by a third step and showed its importance through a numerical study. We concluded that the heuristic works faster and delivers better controls compared to an MIQP solver restricted to one hour computing time. Future work includes the comparison with different models and other heuristics in order to further increase the quality of the control and decrease the computing time. A combination of the three-step heuristic and the space mapping concept[3, 4] is conceivable.

Acknowledgments

S. Göttlich is partly funded by the DFG project GO 1920/4-1. A. Potschka acknowledges funding by the German Federal Ministry of Education and Research (BMBF) program “Mathematics for Innovations in Industry and Service”, grant n^o 05M2016-MoPhaPro and by the European Research Council Adv. Inv. Grant MOBOCON 291 458.

References

[1] I. P. Androulakis, C. D. Maranas, and C. A. Floudas, *αBB: a global optimization method for general constrained nonconvex problems*, J. Global Optim., 7 (1995), pp. 337–363. State of the art in global optimization: computational methods and applications (Princeton, NJ, 1995).

[2] L. Bahiense, G. Oliveira, M. Pereira, and S. Granville, *A Mixed Integer Disjunctive Model for Transmission Network Expansion*, IEEE Transaction on Power Systems, 16 (2001), pp. 560–565.

[3] J. Bandler, R. Biernaki, S. Chen, P. Grobelny, and R. Hemmers, *Space mapping technique for electromagnetic optimization*, IEEE Trans. Microwave Theory Tech., 42 (1994), pp. 2536–2544.

[4] J. Bandler, R. Biernaki, S. Chen, R. Hemmers, and K. Madsen, *Electromagnetic optimization exploiting aggressive space mapping*, IEEE Trans. Microwave Theory Tech., 43 (1994), pp. 2874–2882.

[5] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan, *Mixed-integer nonlinear optimization*, Acta Numer., 22 (2013), pp. 1–131.

[6] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, *Branching and bounds tightening techniques for non-convex MINLP*, Optim. Methods Softw., 24 (2009), pp. 597–634.

[7] J. Bergen and V. Vittal, *Power Systems Analysis*, Prentice Hall, Upper Saddle River, New Jersey, second ed., 2000.

[8] T. Berthold and A. M. Gleixner, *Undercover: a primal MINLP heuristic exploring a largest sub-MIP*, Math. Program., 144 (2014), pp. 315–346.

[9] D. Bienstock, M. Chertkov, and S. Harnett, *Chance-constrained optimal power flow: risk-aware network control under uncertainty*, SIAM Rev., 56 (2014), pp. 461–495.

[10] P. Bonami, *Lift-and-project cuts for mixed integer convex programs*, Integer Programming and Combinatorial Optimization, 6655 (2011), pp. 52–64.

[11] P. Bonami, L. T. Biegler, A. R. Conn, G. Cornuéjols, I. E. Grossmann, C. D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter, *An algorithmic framework for convex mixed integer nonlinear programs*, Discrete Optim., 5 (2008), pp. 186–204.

[12] P. Bonami, G. Cornuéjols, A. Lodi, and F. Margot, *A Feasibility Pump for mixed integer nonlinear programs*, Math. Program., 119 (2009), pp. 331–352.

[13] P. Bonami, J. Lee, S. Leyffer, and A. Wächter, *More branch-and-bound experiments in convex nonlinear integer programming*, Preprint ANL/MCS-P1949-0911, Argonne National Laboratory, Mathematics and Computer Science Division, (2011).

[14] J. Bonnans, *Mathematical study of very high voltage power networks I. The optimal DC power flow problem*, SIAM J. Optim., 7 (1997), pp. 979–990.

[15] R. M. Colombo, G. Guerra, M. Herty, and V. Schleper, *Optimal control in networks of pipes and canals*, SIAM J. Control Optim., 48 (2009), pp. 2032–2050.

[16] E. da Silva, H. Gil, and J. Areiza, *Transmission network expansion planning under an improved genetic algorithm*, IEEE Transaction on Power Systems, 15 (2000), pp. 1168–1175.

[17] R. J. Dakin, *A tree-search algorithm for mixed integer programming problems*, Comput. J., 8 (1965), pp. 250–255.

[18] V. Donde, V. Lopez, B. Lesieutre, A. Pinar, C. Yang, and J. Meza, *Identification of severe multiple contingencies in electric power networks*, In Proceedings 37th North American Power Symposium, LBNL-57994 (2005).

[19] M. Duran and I. Grossmann, *An outer-approximation algorithm for a class of mixed-integer nonlinear programs*, Mathematical Programming, 36 (1986), pp. 307–339.

[20] E. Fisher, R. O'Neil, and M. Ferris, *Optimal transmission switching*, IEEE Transactions on Power Systems, 23 (2008), pp. 1346–1355.

[21] R. Fletcher and S. Leyffer, *Solving mixed integer nonlinear programs by outer approximation*, Math. Programming, 66 (1994), pp. 327–349.

[22] R. Freeman and A. Karbowiak, *An investigation of nonlinear transmission lines and shock waves*, Journal of Physics D, 10 (1977), p. 633.

[23] B. Geissler, A. Martin, A. Morsi, and L. Schewe, *Using piecewise linear functions for solving MINLPs*, in Mixed integer nonlinear programming, vol. 154 of IMA Vol. Math. Appl., Springer, New York, 2012, pp. 287–314.

[24] A. M. Geoffrion, *Generalized Benders decomposition*, J. Optimization Theory Appl., 10 (1972), pp. 237–260.

[25] S. Göttlich, M. Herty, and P. Schillen, *Electric transmission lines: control and numerical discretization*, Optimal Control Appl. Methods, 37 (2016), pp. 980–995.

[26] S. Göttlich, A. Potschka, and U. Ziegler, *Partial outer convexification for traffic light optimization in road networks*, SIAM J. Sci. Comput., 39 (2017), pp. B53–B75.

[27] S. Göttlich and C. Teuber, *Space mapping techniques for the optimal inflow control of transmission lines*, Optimization Methods and Software, (2017), pp. 1–20.

[28] S. Göttlich and U. Ziegler, *Traffic light control: a case study*, Discrete Contin. Dyn. Syst. Ser. S, 7 (2014), pp. 483–501.

[29] J.-P. Goux and S. Leyffer, *Solving large MINLPs on computational grids*, Optim. Eng., 3 (2002), pp. 327–346. Special issue on mixed-integer programming and its applications to engineering.

[30] G. Gross and F. D. Galiana, *Short-term load forecasting*, Proceedings of the IEEE, 75 (1987), pp. 1558–1573.

[31] I. E. Grossmann, *Review of nonlinear mixed-integer and disjunctive programming techniques*, Optim. Eng., 3 (2002), pp. 227–252. Special issue on mixed-integer programming and its applications to engineering.

[32] F. M. Hante, *Relaxation methods for hyperbolic PDE mixed-integer optimal control problems*, Optimal Control Appl. Methods, (2017).

[33] F. M. Hante and S. Sager, *Relaxation methods for mixed-integer optimal control of partial differential equations*, Comput. Optim. Appl., 55 (2013), pp. 197–225.

[34] Y.-L. Jiang, *Mathematical modelling on RLCG transmission lines*, Nonlinear Anal. Model. Control, 10 (2005), pp. 137–149.

[35] M. N. Jung, G. Reinelt, and S. Sager, *The Lagrangian relaxation for the combinatorial integral approximation problem*, Optim. Methods Softw., 30 (2015), pp. 54–80.

[36] C. Kirches, S. Sager, H. G. Bock, and J. P. Schlöder, *Time-optimal control of automobile test drives with gear shifts*, Optimal Control Appl. Methods, 31 (2010), pp. 137–153.

[37] P. Kundur, *Power System Stability and Control*, McGraw-Hill, New York, 1994.

[38] R. Lasseter, *MicroGrids*, IEEE Power Engineering Society Winter Meeting, 1 (2002), pp. 305–308.

[39] S. Leyffer, J. Linderoth, J. Luedtke, A. Miller, and T. Munson, *Applications and algorithms for mixed integer nonlinear programming*, Journal of Physics: Conference Series, 180 (2009).

[40] J. Linderoth and M. Savelsbergh, *A computational study of search strategies in mixed integer programming*, INFORMS Journal on Computing, 11 (1999), pp. 173–187.

[41] C. Liu, J. Wang, and J. Ostrowski, *Static Switching Security in Multi-Period Transmission Switching*, IEEE Transaction on Power Systems, 27 (2012), pp. 1850–1858.

[42] R. Misener and C. A. Floudas, *GloMIQO: global mixed-integer quadratic optimizer*, J. Global Optim., 57 (2013), pp. 3–50.

[43] G. Nannicini and P. Belotti, *Rounding-based heuristics for nonconvex MINLPs*, Math. Program. Comput., 4 (2012), pp. 1–31.

[44] J. Ostrowski and J. Wang, *Network reduction in the Transmission-Constrained Unit Commitment problem*, Computers and Industrial Engineering, 63 (2012), pp. 702–707.

[45] A. Papalexopoulos, S. Hao, and T. Peng, *An implementation of a neural network based load forecasting model for the EMS*, IEEE Transactions on Power Systems, 9 (1994), pp. 1956–1962.

[46] J. Pecas Lopes, C. Moreira, and A. Madureira, *Defining Control Strategies for MicroGrids Islanded Operation*, IEEE Transaction on Power Systems, 21 (2006), pp. 916–924.

[47] R. Romero and A. Monticelli, *A hierarchical decomposition approach for transmission network expansion planning*, IEEE Transaction on Power Systems, 9 (1994), pp. 373–380.

[48] S. Sager, *Numerical methods for mixed-integer optimal control problems*, Der andere Verlag Tönning, Lübeck, Marburg, 2005.

[49] ——, *Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control*, Journal of Process Control, 19 (2009), pp. 1238–1247.

[50] S. Sager, H. G. Bock, and M. Diehl, *The integer approximation error in mixed-integer optimal control*, Math. Program., 133 (2012), pp. 1–23.

[51] S. Sager, H. G. Bock, and G. Reinelt, *Direct methods with maximal lower bound for mixed-integer optimal control problems*, Math. Program., 118 (2009), pp. 109–149.

[52] S. Sager, M. Jung, and C. Kirches, *Combinatorial integral approximation*, Math. Methods Oper. Res., 73 (2011), pp. 363–380.

[53] M. W. P. Savelsbergh, *Preprocessing and probing techniques for mixed integer programming problems*, ORSA J. Comput., 6 (1994), pp. 445–454.

[54] A. Soares, A. Gomes, C. Henggeler-Antunes, and H. Cardoso, *Domestic load scheduling using genetic algorithms*, Lecture Notes in Computer Science, Springer, 7835 (2013), pp. 142–151.

[55] R. Stubbs and S. Mehrotra, *A branch-and-cut method for 0-1 mixed convex programming*, Mathematical Programming, 86 (1999), pp. 515–532.

[56] T. J. Van Roy, *Cross decomposition for mixed integer programming*, Math. Programming, 25 (1983), pp. 46–63.

[57] J. P. Vielma, S. Ahmed, and G. Nemhauser, *Mixed-integer models for nonseparable piecewise-linear optimization: unifying framework and extensions*, Oper. Res., 58 (2010), pp. 303–315.

[58] J. P. Vielma and G. L. Nemhauser, *Modeling disjunctive constraints with a logarithmic number of binary variables and constraints*, Math. Program., 128 (2011), pp. 49–72.