Scalable Primal Decomposition Schemes for Large-Scale Infrastructure Networks

Alexander Engelmann, Sungho Shin, François Pacaud, Victor Zavala

Scalable Primal Decomposition Schemes for Large-Scale Infrastructure Networks

Alexander Engelmann, Sungho Shin,
François Pacaud, Victor M. Zavala*

December 22, 2022

The real-time operation of large-scale infrastructure networks requires scalable optimization capabilities. Decomposition schemes can help achieve scalability; classical decomposition approaches such as the alternating direction method of multipliers (ADMM) and distributed Newtons schemes, however, often either suffer from slow convergence or might require high degrees of communication. In this work, we present new primal decomposition schemes for solving large-scale, strongly convex QPs. These approaches have global convergence guarantees and require limited communication. We benchmark their performance against the off-the-shelf interior-point method Ipopt and against ADMM on infrastructure networks that contain up to 300,000 decision variables and constraints. Overall, we find that the proposed approaches solve problems as fast as Ipopt but with reduced communication. Moreover, we find that the proposed schemes achieve higher accuracy than ADMM approaches.

1. Introduction

The operation of infrastructure networks such as power systems, district heating grids or gas networks is challenging. In many cases, these networks are large and composed of many complex subsystems such as lower-level networks or buildings. Operation is often based on numerical optimization due to its flexibility and recent advances in solver development, which allows to solve large-scale problems quickly and to a high accuracy. For large networks, however, a centralized solution is often not desirable since, a), the

*AE is with TU Dortmund University, Dortmund, Germany (alexander.engelmann@ieee.org). SS is with Argonne National Laboratory, Lemont, IL, USA (sshin@anl.gov). FP is with Centre Mathématiques et Systèmes, Mines Paris-PSL, Paris, France (francois.pacaud@minesparis.psl.eu). VZ is with University of Wisconsin-Madison, Madison, WI, USA and Argonne National Laboratory, Lemont, IL, USA (victor.zavala@wisc.edu).

problem becomes computationally challenging, even with state-of-the-art solvers; b), information collection in a central entity should be avoided due to confidentiality and privacy concerns, and, c), the responsibility for operation and updates in modeling should stay mainly in the subsystems.

One line of research addresses the above challenges via aggregation. Here, the idea is to simplify the subproblems by projecting the constraint set on the coupling variables of the infrastructure network. Examples for this can be found for power systems [1, 2]. A drawback of this approach is a loss of optimality. Moreover, aggregation is often not straightforward, feasibility is hard to guarantee and disaggregation requires solving additional local optimization problems.

A second line of research is based on distributed optimization. Prominent approaches are duality-based first-order algorithms such as Lagrangian dual decomposition and the Alternating Direction Method of Multipliers (ADMM) [3–5]. Application examples range from the operation of power systems [6, 7], over gas networks [8], district heating systems [9, 10], to water networks [11]. With their at most linear rate of convergence, these approaches often require many iterations to converge even for a modest solution quality. This is often prohibitive for real-time implementation.

Distributed second-order methods exhibit faster convergence. Here, classical approaches aim at decomposing the block-structure of the Karush-Kuhn-Tucker (KKT) system within interior-point algorithms [12, 13] or sequential quadratic programming [14]. Alternative second-order methods based on augmented Lagrangians can be found in [15, 16]. As a drawback, these approaches typically require an expensive central coordination, although it is possible to partially alleviate the computational load by decentralizing the Newton-steps [17–19]. Moreover, ensuring global convergence is difficult in a distributed setting, since this requires monitoring a merit function centrally.

Primal decomposition approaches [20–22] come with the advantage of achieving a high degree of feasibility in a small number of iterations. Here, the idea is to partition the problem in a hierarchical fashion, i.e. to construct lower-level problems coordinated by one upper-level problem, where the upper-level problem considers the lower-level problems by their optimal value functions. Primal decomposition has been very successful in solving large-scale problems from chemical engineering [23, 24] and some of the largest Quadratic Programs (QPs) and Nonlinear Programs (NLPs) from power systems [25–27]. Moreover, these approaches allow to use specialized, domain-specific solvers to solve the subproblems and the master problem efficiently [21].

In this work, we propose two primal decomposition schemes for solving large-scale strongly convex QPs, with global convergence guarantees. Both methods rely respectively on augmented Lagrangians and exact $\ell 1$-penalties for ensuring feasibility in the subproblems. Whereas similar $\ell 1$-penalty based approaches have been proposed in previous works [21], the augmented-Lagrangian framework is new to the best of our knowledge. We show that the augmented Lagrangian formulation exhibits improved performance compared to the $\ell 1$ formulation. Moreover, we demonstrate that the algorithms are faster than off-the-shelf interior-point solvers. We benchmark our algorithms against a distributed ADMM and the nonlinear solver `Ipopt` with sparse and parallel linear algebra. As benchmarks, we consider the operation of HVAC systems in a city district with

a variable number of buildings and with up to $300,000$ decision variables and inequality constraints.

Notation

Given $A \in \mathbb{R}^{m\times n}$, $[A]_j$ denotes the jth row of A and $\mathrm{nr}(A) \doteq m$ corresponds to the number of rows of A. The Lagrange multiplier $\lambda \in \mathbb{R}^{n_g}$ associated to the constraint function $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_g}$ is written as $g(x) = 0 \mid \lambda$. Given a vector $v \in \mathbb{R}^n$, $D = \mathrm{diag}(v) \in \mathbb{R}^{n\times n}$ is a matrix with the elements of v on the main diagonal. For a tuple of matrices (A, B), $\mathrm{blkdiag}(A, B)$ denotes their block-diagonal concatenation.

2. Problem Formulation

Many infrastructure network problems can be formulated as strongly convex QPs over a set of subsystems $\mathcal{S} = \{1, \dots, S\}$,

$$\min_{\{x_i\}_{i\in\mathcal{S}}, y} \sum_{i\in\mathcal{S}} \frac{1}{2}\begin{bmatrix} x_i \\ y \end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} \\ H_i^{xy\top} & H_i^{yy} \end{bmatrix} \begin{bmatrix} x_i \\ y \end{bmatrix} + \begin{bmatrix} h_i^x \\ h_i^y \end{bmatrix}^\top \begin{bmatrix} x_i \\ y \end{bmatrix} \tag{1a}$$

$$\text{subject to} \quad \begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & y^\top \end{bmatrix}^\top - b_i = 0, \quad i \in \mathcal{S}, \tag{1b}$$

$$\begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & y^\top \end{bmatrix}^\top - d_i \le 0, \quad i \in \mathcal{S}, \tag{1c}$$

$$A^y y - b^y = 0, \quad B^y y - d^y \le 0. \tag{1d}$$

Here, the global decision variable vector $x = [x_1, \dots, x_S]^\top$ is composed of local decision variables $x_i \in \mathbb{R}^{n_{xi}}$, where each x_i belongs to one subsystem $i \in \mathcal{S}$. The decision variables $y \in \mathbb{R}^{n_y}$ are "global" in the sense that they belong to the interconnecting infrastructure network, described by the constraints (1d). Each coefficient matrix/vector in the objective (1a) and the constraints (1b), (1c) belongs to one $i \in \mathcal{S}$.

3. Primal Decomposition Schemes

In contrast to duality-based techniques such as ADMM or dual decomposition, primal decomposition decomposes entirely in the primal space, i.e. no dual variables are updated in the solution process. The main idea here is to replace the subproblems in (1) by their optimal value functions. Specifically, one reformulates (1) as

$$\min_y \sum_{i\in\mathcal{S}} \phi_i(y), \quad \text{s.t.} \quad A^y y - b^y = 0,\ B^y y - d^y \le 0, \tag{2}$$

where for all $i \in \mathcal{S}$, the value function ϕ_i is defined as

$$\phi_i(y) \doteq \min_{x_i} \frac{1}{2}\begin{bmatrix} x_i \\ y \end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} \\ H_i^{xy\top} & H_i^{yy} \end{bmatrix} \begin{bmatrix} x_i \\ y \end{bmatrix} + \begin{bmatrix} h_i^x \\ h_i^y \end{bmatrix}^\top \begin{bmatrix} x_i \\ y \end{bmatrix} \tag{3a}$$

$$\text{subject to} \quad \begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & y^\top \end{bmatrix}^\top - b_i = 0,\ i \in \mathcal{S}, \tag{3b}$$

$$\begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & y^\top \end{bmatrix}^\top - d_i \le 0,\ i \in \mathcal{S}. \tag{3c}$$

The key idea is to apply standard algorithms for solving (2) by optimizing only with respect to the coupling variables y. Doing so can lead to enhanced robustness, as the complexity of the subproblems is not exposed to the algorithm solving (2).

Algorithms for solving (2) typically require first-order and possibly second-order derivatives of all $\{\phi_i\}_{i\in\mathcal{S}}$. Since all $\{\phi_i\}_{i\in\mathcal{S}}$ are non-smooth because of the inequality constraints, one typically relies on smooth reformulations. Inspired by interior-point methods [21], we introduce log-barrier functions and slack variables $s_i \in \mathbb{R}^{n_{ii}}$, which approximate (3) by

$$\Phi_i^\delta(y) \doteq \min_{x_i,s_i} \frac{1}{2}\begin{bmatrix}x_i\\y\end{bmatrix}^\top \begin{bmatrix}H_i^{xx} & H_i^{xy}\\ H_i^{xy\top} & H_i^{yy}\end{bmatrix}\begin{bmatrix}x_i\\y\end{bmatrix} + \begin{bmatrix}h_i^x\\h_i^y\end{bmatrix}^\top \begin{bmatrix}x_i\\y\end{bmatrix} - \delta \mathbb{1}^\top \ln(s) \tag{4a}$$

$$\text{subject to} \quad \begin{bmatrix}A_i^x & A_i^y\end{bmatrix}\begin{bmatrix}x_i^\top & y^\top\end{bmatrix}^\top - b_i = 0,\ i\in\mathcal{S}, \tag{4b}$$

$$\begin{bmatrix}B_i^x & B_i^y\end{bmatrix}\begin{bmatrix}x_i^\top & y^\top\end{bmatrix}^\top - d_i + s_i = 0,\ i\in\mathcal{S}, \tag{4c}$$

where $\delta \in \mathbb{R}_+$ is a barrier parameter, $\mathbb{1} \doteq [1,\dots,1]^\top$, and the $\ln(\cdot)$ is evaluated component-wise. Note that $\lim_{\delta\to 0}\Phi_i^\delta(y) = \phi_i(y)$, and that Φ_i^δ is smooth[1]. A basic primal decomposition strategy with smoothing is summarized in Algorithm 1.

Algorithm 1: A basic primal decomposition scheme.

Initialize y^0, δ^0.
while not terminated **do**
  1) Solve (2) for $\phi_i \equiv \Phi_i^\delta$ with a NLP solver; in case the NLP solver calls $(\nabla_y\Phi_i^\delta, \nabla_y^2\Phi_i^\delta)$, compute them locally for all $i\in\mathcal{S}$.
  3) Decrease δ.
end
Return y^k, $\{x_i^k\}_{i\in\mathcal{S}}$.

Dealing With Infeasibility

An issue in Algorithm 1 is that the subproblems (3) may be infeasible for a given y. One way of circumventing this is to introduce auxiliary variables $z_i \in \mathbb{R}^{n_y}$ and to use relaxation techniques either based on augmented Lagrangians or on exact $\ell 1$-penalties. Consider a set of auxiliary variables $\{z_i\}_{i\in\mathcal{S}}$ and introduce additional constraints $z_i = y$

[1] Under standard regularity assumptions [22, A1-C1].

for all $i \in \mathcal{S}$. Then, we reformulate (4) equivalently as [2]

$$\Phi_i^\delta(y) = \min_{x_i,s_i,z_i} \frac{1}{2}\begin{bmatrix} x_i \\ y \end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} \\ H_i^{xy\top} & H_i^{yy} \end{bmatrix} \begin{bmatrix} x_i \\ y \end{bmatrix} + \begin{bmatrix} h_i^x \\ h_i^y \end{bmatrix}^\top \begin{bmatrix} x_i \\ y \end{bmatrix} - \delta \mathbb{1}^\top \ln(s) \tag{5a}$$

$$\text{subject to } \begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - b_i = 0,\ i \in \mathcal{S}, \tag{5b}$$

$$\begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - d_i + s_i = 0,\ i \in \mathcal{S}, \tag{5c}$$

$$z_i = y,\ i \in \mathcal{S}, \tag{5d}$$

which can still be infeasible, but paves the way for augmented Lagrangian and exact $\ell 1$ relaxations.

Augmented Lagrangian Relaxation

A simple way of making (5) feasible for all y is to relax (5d) via a quadratic penalty [21]. However, in this case, large penalty parameters might lead to numerical difficulties and feasibility can in general not be guaranteed for a finite penalty parameter. Hence, we use an Augmented Lagrangian (AL) approach to solve (2) for late outer iterations with a constant barrier parameter δ. Assigning the Lagrange multiplier λ_i to (5d), we relax (5d) in an Augmented Lagrangian fashion by adding the terms $\lambda_i^{k\top}(y - z_i) + \frac{\rho}{2}\|y - z_i\|_2^2$ to the objective:

$$\begin{aligned} \Phi_i^{\delta,\rho}(y, \lambda_i^k) \doteq \min_{x_i,s_i,z_i} \frac{1}{2}\begin{bmatrix} x_i \\ y \\ z_i \end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} & 0 \\ H_i^{xy\top} & H_i^{yy} + \rho I & -\rho I \\ 0 & -\rho I & \rho I \end{bmatrix} \begin{bmatrix} x_i \\ y \\ z_i \end{bmatrix} + \begin{bmatrix} h_i^x \\ h_i^y + \lambda_i^k \\ -\lambda_i^k \end{bmatrix}^\top \begin{bmatrix} x_i \\ y \\ z_i \end{bmatrix} - \delta \mathbb{1}^\top \ln(s_i) \\ \text{subject to } \begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - b_i = 0, \quad \mid \gamma_i \\ \begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top + s_i - d_i = 0, \quad \mid \mu_i. \end{aligned} \tag{6}$$

Here, (γ_i, μ_i) are Lagrange multipliers corresponding to the constraints in the same line. Note that the solution of (5) can be forced to be arbitrarily close to that of (6) by letting $\rho \to \infty$, if (5) is feasible for y. In addition, if one has a good Lagrange multiplier estimate $\lambda^k \approx \lambda^\star$, (5) and (6) are equivalent for a finite $\rho < \infty$ [28, Sec. 3.2.1], [29, Thm. 17.5]. We will exploit this fact in the following.

Primal decomposition based on the augmented Lagrangian works as follows: In phase 1, the barrier parameter δ and the penalty parameter ρ are increased/decreased simultaneously. In phase 2, when δ/ρ are sufficiently small/large, both are held constant and a standard augmented Lagrangian algorithm is applied to the resulting optimization problem to obtain feasibility in (5d). We use the standard first-order update rule from augmented Lagrangian algorithms [29, Chap. 17.3]

$$\lambda_i^{k+1} = \lambda_i^k + \rho(y^k - z_i^k). \tag{7}$$

The resulting scheme is summarized in Algorithm 2.

[2] Observe that we have replaced y by z_i in the constraints here but not in the objective. Exchanging y in the objective is possible but might lead to a different numerical behavior.

Algorithm 2: AL-based primal decomposition.

Initialize y^0, δ^0, ρ^0; $\lambda_i = 0, i \in \mathcal{S}$.
while phase 1 **do**
  1) Solve (2) for $\phi_i \equiv \Phi_i^{\delta,\rho}$ with a NLP solver; in case the NLP solver calls $(\nabla_y \Phi_i^{\delta,\rho}, \nabla^2_{yy} \Phi_i^{\delta,\rho})$, compute them locally for all $i \in \mathcal{S}$.
  3) Decrease δ, increase ρ.
end
while phase 2 **do**
  1) Solve (2) as in phase 1.
  2) Update λ_i^k according to (7).
end
Return y^k, $\{x_i^k\}_{i\in\mathcal{S}}$.

$\ell 1$-penalty relaxation

A second variant to ensure feasibility is to relax (5d) via an $\ell 1$-penalty function. This has the advantage that it is exact also for a finite penalty parameter $\bar{\lambda} \in \mathbb{R}_+$ without the need for Lagrange-multiplier estimation. By doing so, the objective becomes non-smooth. However, the non-smoothness can be eliminated by using an elastic relaxation [29, p.535]: the $\ell 1$-penalty $\min_{y,z_i} \|y - z_i\|_1$ is reformulated by introducing two non-negative auxiliary variables $v_i, w_i \in \mathbb{R}_+^{n_y}$ as $\min_{y,z_i,v_i,w_i} v_i + w_i$ subject to $y - z_i = v_i - w_i$. The corresponding reformulation of (5) reads

$$\begin{aligned}
\Phi_i^{\delta,\bar{\lambda}}(y) \doteq \min_{\substack{x_i,s_i,z_i\\ v_i,w_i}} \; & \frac{1}{2}\begin{bmatrix} x_i \\ z_i \end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} \\ H_i^{xy\top} & H_i^{yy} \end{bmatrix} \begin{bmatrix} x_i \\ z_i \end{bmatrix} + \begin{bmatrix} h_i^x \\ h_i^y \end{bmatrix}^\top \begin{bmatrix} x_i \\ z_i \end{bmatrix} \\
& + \bar{\lambda} \mathbb{1}^\top (v_i + w_i) - \delta(\mathbb{1}^\top \ln(s_i) + \mathbb{1}^\top \ln(v_i) + \mathbb{1}^\top \ln(w_i)) \quad (8) \\
\text{subject to} \quad & \begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - b_i = 0, \quad \mid \gamma_i, \\
& \begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top + s_i - d_i = 0, \quad \mid \mu_i, \\
& y - z_i - v_i + w_i = 0, \quad \mid \chi_i,
\end{aligned}$$

where the bounds $(v_i, w_i) \geq 0$ are replaced by log-barrier functions. If one chooses $\bar{\lambda} < \infty$ large enough, (5) and (8) are equivalent [29, Thm 17.3].

Primal decomposition based on the $\ell 1$-penalty solves (2) using $\Phi_i^{\delta,\bar{\lambda}}$ with a fixed $\bar{\lambda}$ larger than a certain threshold and decreases the barrier parameter δ during the iterations. The overall algorithm is summarized in Algorithm 3.

Algorithm 3: $\ell 1$-based primal decomposition.

Initialize $y^0, \delta^0, \bar{\lambda}$ large enough.
while not terminated **do**
  1) Solve (2) for $\phi_i \equiv \Phi_i^{\delta,\bar{\lambda}}$ with a NLP solver; in case the NLP solver calls $(\nabla_y \phi_i, \nabla^2_{yy}\phi_i)$, compute them locally for all $i \in \mathcal{S}$.
  3) Decrease δ.
end
Return y^k, $\{x_i^k\}_{i\in\mathcal{S}}$.

4. Computing sensitivities

Next, we show how to compute $\nabla_y \Phi_i^\delta$ and $\nabla^2_{yy}\Phi_i^\delta$ under standard regularity assumptions based on the implicit function theorem [22]. Reformulate (4) by

$$\begin{aligned} \Phi_i^\delta(y) = \min_{x_i, s_i} \; & f_i^\delta(x_i, s_i; y) \\ \text{subject to} \quad & g_i(x_i; y) = 0 \mid \gamma_i, \quad h_i(x_i; y) + s_i = 0 \mid \mu_i, \end{aligned} \tag{9}$$

where f_i^δ is defined by (4a), and g_i and h_i are defined by (4b), (4c). Define the Lagrangian to (9),

$$L_i^\delta(x_i, s_i, \gamma_i, \mu_i; y) \doteq f_i^\delta(x_i, s_i; y) + \gamma_i^\top g_i(x_i; y) + \mu_i^\top (h_i(x_i; y) + s_i).$$

Assume that (9) is feasible for a given y and that regularity conditions such as LICQ, SOSC and SCC hold, cf. [22, Ass. 1-4]. Then, the KKT conditions to (9) form an implicit function in form of $F_i^\delta(x_i^\star, s_i^\star, \gamma_i^\star, \mu_i^\star; y) = 0$, where the superscript $(\cdot)^\star$ indicates a KKT stationary point. Thus, by the implicit function theorem, there exist a neighborhood around y for which there exists functions $p_i^\star(y) \doteq (x_i^\star(y), s_i^\star(y), \gamma_i^\star(y), \mu_i^\star(y))$ such that $F_i^\delta(p_i^\star(y); y) = 0$. Hence, we can rewrite (9) as $\Phi_i^\delta(y) = f_i(x_i^\star(y), s_i^\star(y); y) = L_i(p_i^\star(y); y)$ since $p_i^\star(y)$ is feasible.

Applying the total derivative and the chain rule yields

$$\nabla_y \Phi_i^\delta(y) = \nabla_y L_i^\delta(p_i^\star(y); y) + \nabla_{p_i^\star} L_i^\delta(p_i^\star(y); y) \nabla_y p_i^\star(y).$$

By the KKT conditions, we have that $\nabla_{p_i^\star} L_i^\delta(p_i^\star(y); y) = 0$ and thus

$$\nabla_y \Phi_i^\delta(y) = \nabla_y L_i^\delta(p_i^\star(y); y). \tag{10}$$

Again by the total derivative, the Hessian can be computed by

$$\nabla^2_{yy} \Phi_i^\delta(y) = \nabla^2_{yy} L_i(p_i^\star(y); y) + \nabla^2_{y p_i^\star} L_i(p_i^\star(y); y)\, \nabla_y p_i^\star(y). \tag{11}$$

It remains to derive an expression for $\nabla_y p_i^\star(y)$. The KKT conditions of (9) read

$$F_i^\delta(x_i^\star, s_i^\star, \gamma_i^\star, \mu_i^\star; y) = \begin{bmatrix} \nabla_{x_i} f_i(x_i^\star, y) + \nabla_{x_i} g_i(x_i^\star, y)\gamma_i^\star + \nabla_{x_i} h_i(x_i^\star, y)\mu_i^\star \\ -\delta S_i^{\star -1}\mathbb{1} + \mu_i^\star \\ g_i(x_i^\star, y) \\ h_i(x_i^\star, y) + s_i^\star \end{bmatrix} \stackrel{!}{=} 0,$$

where $S_i^\star = \operatorname{diag}(s_i^\star)$. By the total differential and the chain rule we have $\nabla_y F_i^\delta(p_i^\star(y), y) + \nabla_{p_i^\star} F_i^\delta(p_i^\star(y), y)\nabla_y p_i^\star(y) = 0$. Hence, we can compute the Jacobian $\nabla_y p_i^\star(y)$ by solving the system of linear equations

$$\left(\nabla_{p_i^\star} F_i^\delta(p_i^\star(y), y)\right) \nabla_y p_i^\star(y) = -\nabla_y F_i^\delta(p_i^\star(y), y). \tag{12}$$

Observe that (12) is a system of linear equations with multiple right-hand sides. In summary, we can compute $\nabla_{yy}^2 \Phi_i^\delta(y)$ locally for each $i \in \mathcal{S}$ by combining (11) and (12). The corresponding formulas for the gradient and the Hessian of $\Phi_i^{\delta,\rho}$ and $\Phi_i^{\delta,\bar{\lambda}}$ from (6) and (8), i.e. of the AL relaxation and the $\ell 1$ relaxation (9) are given in Appendix A.

5. Solving the Master Problem and Globalization

An important question is how to solve the master problem (2) for different variants of ϕ_i. In general, this can be done by any sensitivity-based NLP solver. We proceed by showing how to obtain a simple globalized version of Algorithm 1 based on a line-search scheme; here, the idea is to show global convergence for the relaxed problem (2) with $\phi_i \in \{\Phi_i^{\delta,\rho}, \Phi_i^{\delta,\bar{\lambda}}\}$ for fixed penalty and barrier parameters. This leads to converge of a solution to the original problem (1) by standard results from penalty and barrier methods [29, Thms. 17.1, 17.6].

Define the objective of (2), $\psi(y) \doteq \sum_{i\in\mathcal{S}} \phi_i(y)$, as a global merit function, where $\phi_i \in \{\Phi_i^{\delta,\rho}, \Phi_i^{\delta,\bar{\lambda}}\}$. The basic idea is to employ a Sequential Quadratic Programming (SQP) scheme, where we ensure a sufficient decrease in ψ at each step via the Armijo condition. The overall algorithm is summarized in Algorithm 4. Similar to the general primal decomposition scheme from Algorithm 1, the master problem solver evaluates the sensitivities $(\nabla_y \phi_i, \nabla_{yy}^2 \phi_i)$ in step (i), in order to construct a quadratic approximation of (2) in step (ii). Solving this approximation yields a search direction Δy. The stepsize α is updated with a backtracking line-search with the Armijo condition as termination criterion.

Global Convergence

We now establish global convergence[3] of Algorithm 4 to a minimizer of the relaxed problems (2) for $\phi_i \in \{\Phi_i^{\delta,\rho}, \Phi_i^{\delta,\bar{\lambda}}\}$. Assume that the following regularity assumptions

[3] We use the definition of global convergence in the context of NLPs, i.e. the convergence to a KKT point from an arbitrary initialization [29, Chap. 3]. However, since problems (1) and (2) are strongly convex and we assume regularity in Assumption 1, every KKT point is also a global minimizer.

Algorithm 4: A simple master problem solver.

Initialize $y^0, \epsilon, \delta, \zeta \in (0,1), \rho$ or $\bar{\lambda}, \phi_i \in \{\Phi_i^{\delta,\rho}, \Phi_i^{\delta,\bar{\lambda}}\}$.
while $\|\Delta y\| > 0$ **do**
  (i) compute $(\nabla_y \phi_i, \nabla^2_{yy}\phi_i)$ locally for all $i \in \mathcal{S}$;
  (ii) solve the coordination problem

$$\min_{\Delta y} \sum_{i\in\mathcal{S}} \frac{1}{2}\Delta y \nabla^2_{yy}\phi_i(y)\Delta y^\top + \nabla_y \phi_i(y)^\top \Delta y \qquad (13)$$
$$\text{s.t. } A^y(y+\Delta y) - b^y = 0, B^y(y+\Delta y) - b^y \leq 0.$$

  $\alpha = 1$
  while $\psi(y) - \psi(y+\alpha\Delta y) \geq -\sigma\alpha\nabla_y\psi(y)^\top\Delta y$ **do**
    $\alpha \leftarrow \zeta\alpha$
  end
  $y \leftarrow y + \alpha\Delta y$
end
Return y, $\{x_i\}_{i\in\mathcal{S}}$.

hold at the optimal solution at $p^\star \doteq [p_i^\star(y^\star)]_{i\in\mathcal{S}}$.

Assumption 1 (Regularity)**.** *Assume that for all $i \in \mathcal{S}$*

a) $\begin{bmatrix}\Delta x_i^\star \\ \Delta y^\star\end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} \\ H_i^{xy\top} & H_i^{yy}\end{bmatrix}\begin{bmatrix}\Delta x_i^\star \\ \Delta y^\star\end{bmatrix} > 0$, *for all* $\begin{bmatrix}\Delta x_i^\star \\ \Delta y^\star\end{bmatrix} \neq 0$ *with* $\begin{bmatrix}A_i^x & A_i^y\end{bmatrix}\begin{bmatrix}\Delta x_i^\star \\ \Delta y^\star\end{bmatrix} = 0$;

b) $\begin{bmatrix} A_i^x & A_i^y \\ B_i^x & B_i^y\end{bmatrix}$ *has full row rank;*

c) $[\mu_i^\star]_j + \left[[B_i^x \;\; B_i^y][x_i^{\star\top} \; y^{\star\top}]^\top\right]_j \neq 0$, $\forall j = 1, \ldots, \mathrm{nr}(B_i^x)$.

Line-search methods require that the search direction Δy is a descent direction, i.e. $\Delta y \left(\sum_{i\in\mathcal{S}} \nabla_i \phi_i^\delta(y)\right) < 0$. This can be ensured by showing that $\sum_{i\in\mathcal{S}} \nabla^2_{yy}\phi_i^\delta \succ 0$ for all variants of ϕ_i, which we do with the next lemma.

Lemma 1 (Positive definite Hessians)**.** *Let Assumption 1 hold and assume that $(s_i, \mu_i, v_i, w_i) > 0$. Then, a), the Hessian $\nabla^2_{yy}\Phi_i^{\delta,\rho}$ is positive definite for all $\rho > 0$. Moreover, b), $\nabla^2_{yy}\Phi_i^{\delta,\bar{\lambda}}$ is positive definite if $\bar{\lambda}$ is larger than all multipliers associated to the elastic constraints $\bar{\lambda} > \max_j |[\chi_i^\star]_j|$.*

The proof of Lemma 1 is given in Appendix B. Now we are able to show global convergence of Algorithm 4 to the solution of problem (2) with $\phi_i \in \{\Phi_i^{\delta,\rho}, \Phi_i^{\delta,\bar{\lambda}}\}$ for fixed penalty and barrier parameters.

Theorem 1 (Convergence of Algorithm 4 for fixed $\delta, \rho, \bar{\lambda}$)**.** *Consider Algorithm 4 with either fixed* (δ, ρ, λ) *if* $\phi_i = \Phi_i^{\delta,\rho}$ *or with fixed* $(\delta, \bar{\lambda})$ *if* $\phi_i = \Phi_i^{\delta,\bar{\lambda}}$*. Let Assumption 1 and the conditions of Lemma 1 hold. Then, Algorithm 4 converges to the global minimizer of problem* (2) *with* $\phi_i \in \{\Phi_i^{\delta,\rho}, \Phi_i^{\delta,\bar{\lambda}}\}$.

Proof. The unconstrained minimizer to (13), $\Delta y = -\sum_{i\in\mathcal{S}} \nabla^2_{yy}\phi_i^{-1}(y) \sum_{i\in\mathcal{S}} \nabla_y \phi_i(y)$, is a descent direction for the merit function $\psi(\cdot)$ since $\sum_{i\in\mathcal{S}} \nabla_y \phi_i(y)^\top \Delta y < 0$ by the positive definiteness of the Hessians from Lemma 1. Observe that $\Delta y = 0$ is feasible for (13) by feasibility of y. This shows that either $\Delta y = 0$, or Δy is a descent direction. Hence, by [29, Lem 3.1], there exists an $\alpha \in (0, 1]$ such that $\psi(y) - \psi(y + \alpha\Delta y) \geq -\sigma\alpha\nabla_y\psi(y)^\top \Delta y$ is satisfied and thus the inner while loop is well defined. Moreover, by the convergence of line-search methods [29, Thm 3.2], Algorithm 4 will either converge to a stationary point of ψ returning $\Delta y = 0$. Alternatively, if the unconstrained search direction is blocked by the constraints in (13), $\Delta y = 0$ is returned since $\Delta y = 0$ is feasible.

We now show that y is optimal for (2) if $\Delta y = 0$. The KKT conditions associated to (13) read

$$\begin{cases} \displaystyle\sum_{i\in\mathcal{S}} \nabla^2_{yy}\phi_i(y)\Delta y + \nabla_y\phi_i(y)^\top + A^{y\top}\gamma_y + B^{y\top}\mu_y = 0 \\ \qquad A^y(y + \Delta y) - b^y = 0, \quad B^y(y + \Delta y) - b^y \leq 0, \\ \qquad\qquad (B^y(y + \Delta y) - b^y)^\top \mu = 0, \quad \mu \geq 0, \end{cases}$$

which are precisely the KKT conditions for (2) if $\Delta y = 0$. Since (2) is convex and Assumption 1 holds, the assertion follows. □

Combining Theorem 1 with the convergence results for the $\ell 1$-penalty method [29, Them 17.3] or the augmented Lagrangian method [30, Prop 2.7] implies convergence of Algorithm 4 to the minimizer of the original problem (1) for sufficiently large penalties.

Remark 1 (Satisfying the assumptions of Theorem 1)**.** *Observe that the assumptions for Theorem 1 are standard regularity assumptions from nonlinear programming [22, Ass. 1-4]. Moreover,* $(s_i, \mu_i, v_i, w_i) > 0$ *is always ensured when using interior-point solvers for solving* (5) *and* (6) *even in the case of early termination.*

6. Implementation Aspects

Solving Local Subproblems

The evaluation of the sensitivities of $\phi_i \in \{\Phi_i^{\delta,\rho}, \Phi_i^{\delta,\bar{\lambda}}\}$ requires solving local optimization problems (6) or (8) for fixed $\delta, \rho, \bar{\lambda}$. Observe that this can be done using specialized and optimized interior-point solvers, if they allow termination once a certain barrier δ is reached. Moreover, interior-point solvers factorize the KKT matrices $\nabla_{p_i} F_i^\delta$ (cf. (23), (26)) at each inner iteration and these factorizations can be re-used for Hessian computation via (12). Here we provide two variants: our own interior-point QP solver

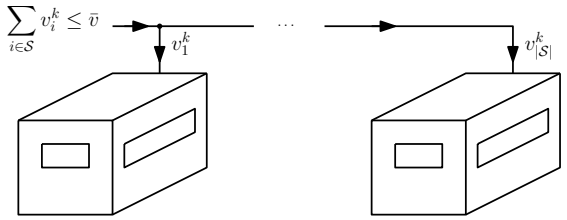

Figure 1: Buildings connected via a network with limited capacity.

based on standard techniques for stepsize selection and barrier parameter decrease [29, Chap. 16.6] and the option to use third-party solvers such as `Ipopt` [31].

In early iterations, it is typically not necessary to solve the local problems to a high precision, since the barrier parameter δ are still large and the penalty parameters $(\rho, \bar{\lambda})$ are still small. Hence, we solve the subproblems to an accuracy measured in the violation of the optimality conditions $\|F_i^\delta(p_i^k, y^k)\|_\infty$ and terminate if $\|F_i^\delta(p_i^k, y^k)\|_\infty < \min(\delta, 1/\rho)$ or $\|F_i^\delta(p_i^k, y^k)\|_\infty < \min(\delta, 1/\bar{\lambda})$. This is inspired by the termination of inexact interior-point methods [32]. Warm-starting the local solves with the solution of the previous iteration reduces computation time significantly.

Numerical Linear Algebra

Efficient numerical linear algebra is crucial for performance. The most expensive steps in terms of memory and CPU time are the factorization of the local KKT matrices and the backsolves in (12). Since we only consider QPs, large parts of the KKT matrices in (12) are constant over the iterations, which can be exploited for pre-computation. Here, we make heavy use of the Schur complement. Details on how to achieve this for the AL formulation are given in Appendix A.

Updating Penalties

We use simple update rules for the penalty parameters. They are

$$\delta^{k+1} = 0.2\delta^k, \quad \bar{\lambda}^{k+1} = 2\bar{\lambda}^k, \quad \rho^{k+1} = 3\rho^k,$$

with initial values $\delta^0 = 0.1, \rho^0 = 10^3$, and $\bar{\lambda}^0 = 100$. After a fixed amount of 8 iterations, phase 2 starts and the above values stay constant.

7. Numerical Case Studies

We consider an optimal control problem for a city district with a scalable number of commercial buildings connected via a electricity grid with limited capacity. The building data is from [33]. We neglect the waterside HVAC system and assume that the buildings are equipped with heat pumps with a constant coefficient of performance.

Building Model

The evolution of the temperature of the mth zone in the ith building reads

$$C_{m,i}(T_{m,i}^{k+1} - T_{m,i}^k) = -H_{m,i}(T_{m,i}^k - T_a^k) - \sum_{n \in \mathcal{Z}_i} G_{mn,i}(T_{m,i}^k - T_{n,i}^k) - \dot{Q}_{m,i}^{ck} + \dot{Q}_{m,i}^{dk}, \quad (14)$$

where at time step k, $T_{m,i}^k$ is the temperature of zone m and T_a^k the ambient temperature, $C_{i,m}$ is the thermal capacity, $H_{m,i}$ and $G_{mn,i}$ are heat transfer coefficients with the ambient and between two zones. Moreover, $(\dot{Q}_{m,i}^{ck}, \dot{Q}_{m,i}^{dk})$ are the controllable/uncontrollable heat influxes from the heat pump and from sources of disturbance such as solar irradiation and occupancy. Eq. (14) can be written in compact form as

$$C_i(T_i^{k+1} - T_i^k) = -H_i(T_i^k - T_a^k) - G_i T_i^k - \dot{Q}_i^{ck} + \dot{Q}_i^{dk},$$

where $T_i^\top \doteq [T_{1,i}, \dots T_{|\mathcal{Z}_i|,i}], \dot{Q}_i^{ck\top} \doteq [\dot{Q}_{1,i}^{ck}, \dots \dot{Q}_{n_z,i}^{ck}]$ and $\dot{Q}_i^{dk\top} \doteq [\dot{Q}_{1,i}^{dk}, \dots \dot{Q}_{n_z,i}^{dk}]$. This yields a state-space model

$$\begin{aligned} T_i^{k+1} \doteq z_i^{k+1} &= \left(I - C_i^{-1}(H_i + G_i)\right) T_i^k - C_i^{-1}\dot{Q}_i^{ck} + C_i^{-1}[H_i \quad I][T_a^{k\top} \quad \dot{Q}_i^{dk\top}]^\top & (15)\\ &\doteq A_i x^k + B_i u_i^k + E_i d_i^k. & (16) \end{aligned}$$

Stacking the above over N time steps yields

$$\begin{aligned} \bar{z}_i &= \begin{bmatrix} 0 & 0 \\ I_{N-1} \otimes A_i & 0 \end{bmatrix} \bar{z}_i + \begin{bmatrix} 0 \\ I_{N-1} \otimes B_i \end{bmatrix} \bar{u}_i + \begin{bmatrix} 0 \\ I_{N-1} \otimes E_i \end{bmatrix} \bar{d}_i + \bar{z}_i^f \\ &\doteq \bar{A}_i \bar{z}_i + \bar{B}_i \bar{u}_i + \bar{E}_i \bar{d}_i + \bar{z}_i^f, \end{aligned}$$

where $\bar{z}_i^\top \doteq [z_i^{0\top}, \dots, z_i^{N\top}]$, $\bar{u}_i^\top \doteq [u_i^{1\top}, \dots, u_i^{N\top}]$, $\bar{d}_i^\top \doteq [d_i^{1\top}, \dots, d_i^{N\top}]$, $\bar{z}_i^{f\top} \doteq [z_i^f, 0^\top, \dots, 0^\top]$ and z_i^f are the initial temperatures. Define the total energy consumption of building $i \in \mathcal{S}$ at time step k by $v_i^k = \mathbb{1}^\top u_i^k$, and $\bar{v}_i^\top \doteq [v_i^{0\top}, \dots, v_i^{N\top}]$. Then, the above is equivalent to

$$\begin{bmatrix} (I - \bar{A}_i) & -\bar{B}_i & 0 \end{bmatrix} \begin{bmatrix} \bar{z}_i^\top & \bar{u}_i^\top & \bar{v}_i^\top \end{bmatrix}^\top = \bar{E}_i \bar{d}_i + \bar{z}_i^f. \quad (17)$$

The grid coupling between all subsystems $i \in \mathcal{S}$ induces an upper-bounded energy supply writing as a global constraint: $\mathbb{1}^\top v_i^k \leq \bar{v}$ for all times k. Moreover, we have local comfort constraints $\underline{T}\mathbb{1} \leq z_i^k \leq \mathbb{1}\bar{T}$.

Optimal Control Problem

Assume that the goal of each building is to minimize its cost of energy consumption over N time steps respecting all constraints. The cost function to buy the power form the utility is given by $f_i^k(u_i^k) \doteq 0.5\, c^k (u_i^k)^2 + g^k u_i^k$, where g^k is a linear and $c^k > 0$ is a (small)

quadratic cost coefficient. This yields a discrete-time Optimal Control Problem (OCP) for building i,

$$\phi_i(v_i) \doteq \min_{\bar{z}_i,\bar{u}_i,\bar{v}_i} \frac{1}{2}\begin{bmatrix}\bar{z}_i\\ \bar{u}_i\\ \bar{v}_i\end{bmatrix}^\top\begin{bmatrix}0 & 0 & 0\\ 0 & cI & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\bar{x}_i\\ \bar{u}_i\\ \bar{v}_i\end{bmatrix}+\begin{bmatrix}0\\ \mathbb{1}^\top\otimes g\\ 0\end{bmatrix}^\top\begin{bmatrix}\bar{z}_i\\ \bar{u}_i\\ \bar{v}_i\end{bmatrix}$$

$$\text{subject to} \quad (17), \begin{bmatrix}I & 0 & 0\\ -I & 0 & 0\end{bmatrix}\begin{bmatrix}\bar{z}_i\\ \bar{u}_i\\ v_i\end{bmatrix} \leq \begin{bmatrix}\bar{T}\mathbb{1}\\ -\underline{T}\mathbb{1}\end{bmatrix},\ \bar{v}_i = I\otimes\mathbb{1}^\top\bar{u}_i.$$

The overall OCP — including global grid constraints — reads

$$\min_{v_1,\ldots,v_{|\mathcal{S}|}} \sum_{i\in\mathcal{S}}\phi_i(v_i), \quad \text{subject to} \quad (\mathbb{1}^\top\otimes I)\,[v_1^\top,\ldots,v_{|\mathcal{S}|}^\top]^\top \leq \bar{v}\,\mathbb{1}. \tag{18}$$

To obtain a problem in form of (1), define $x_i^\top = [\bar{x}_i^\top, \bar{u}_i^\top, \bar{v}_i^\top], i\in\mathcal{S}$, $y^\top = [v_1^\top,\ldots,v_{|\mathcal{S}|}^\top]$, $H_i^x = \text{blkdiag}(0,cI,0), h_i^x = [0, \quad \mathbb{1}^\top\otimes g, 0]^\top, h_i^y = 0, H_i^{xy} = 0, B_i^y = 0, d_i^\top = [\bar{T}\mathbb{1}^\top,\ -\underline{T}\mathbb{1}^\top]$,

$$A_i^x = \begin{bmatrix}(I-\bar{A}_i) & -\bar{B}_i & 0\\ 0 & I\otimes\mathbb{1}^\top & 0\end{bmatrix},\ A_i^y = \begin{bmatrix}0\\ -e_i^\top\otimes I\end{bmatrix},\ B_i^x = \begin{bmatrix}I & 0 & 0\\ -I & 0 & 0\end{bmatrix},$$

where e_i is the ith unit vector, $b_i^\top = \begin{bmatrix}(\bar{E}_i\bar{d}_i + \bar{z}_i^f)^\top & 0\end{bmatrix}$, $A^y = 0$, $b^y = 0$, $B^y = \mathbb{1}^\top\otimes I$, and $d^y = \bar{v}\mathbb{1}$.

Technical Setup

We benchmark our algorithms against ADMM (as one of the most popular algorithms for decomposition) and against `Ipopt` (as one of the most prominent centralized NLP solvers).[4] The particular variant of ADMM is described in Appendix C. We rely on `OSQP` v0.6.2 [34] for solving subproblems and the coordination problem in ADMM. In primal decomposition, we rely on our own interior-point solver for the subproblems and on Algorithm 4 for coordination, where we solve (13) via `Ipopt`.

We perform all simulations on a shared-memory virtual machine with 30 cores and 100GiB memory. The underlying hardware is exclusively used for the case studies. All algorithms are parallelized via Julia multi-threading. `Ipopt` v3.14.4 runs partially in parallel via the parallel multi-frontal linear algebra solver `MUMPS` v5.4.1.

Numerical Results

We compare the numerical performance of all algorithms on OCP (18) for $|\mathcal{S}| \in \{30, 180, 300\}$ buildings. Table 2 shows the corresponding number of local/global decision variables n_x/n_y, the number of local equality/inequality constraints n_e/n_i, and the number of global equality/inequality constraints n_{ey}/n_{iy}. We employ ADMM from Appendix C with penalty parameters $\rho \in \{10^1, 10^2, 10^3, 10^4\}$.

[4]The ADMM-based QP solver `OSQP` solver did not converge for the problems presented here.

Table 1: #decision variables and #constraints for $|\mathcal{S}| \in \{30, 180, 300\}$ buildings.

$\lvert\mathcal{S}\rvert$	n_x	n_y	n_e	n_i	n_{ey}	n_{iy}
30	28,200	690	15,090	28,800	0	1,403
180	169,200	4,140	90,540	172,800	0	8,303
300	282,000	6,900	150,900	288,000	0	13,823

Table 2: Timing and #iter for $|\mathcal{S}| \in \{30, 180, 300\}$ buildings, 30 cores.

$\lvert\mathcal{S}\rvert$	Alg.	AL	$l1$	`Ipopt` (par. LA)	ADMM ($\rho = 10$)	ADMM ($\rho = 100$)
300	t[s]	431.5	-	**386.7**	(522.3)	(611.8)
180		**195.7**	-	218.1	(62.9)	(143.2)
30		**18.1**	270.0	25.5	(20.4)	(16.9)
300	iter.	**13**	-	145	100	
180		**12**	-	141	100	
30		13	**12**	104	100	
	term.	rel. opt. 10^{-4} infeas 10^{-5}		optimality	fixed #iter	

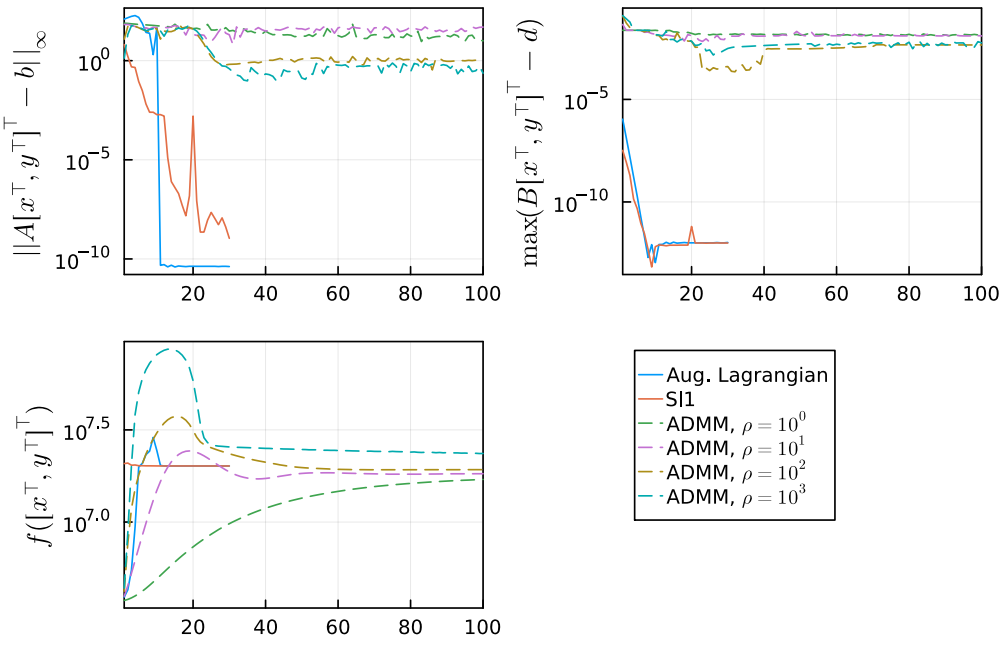

Figure 2: Convergence for $|\mathcal{S}| = 30$ buildings.

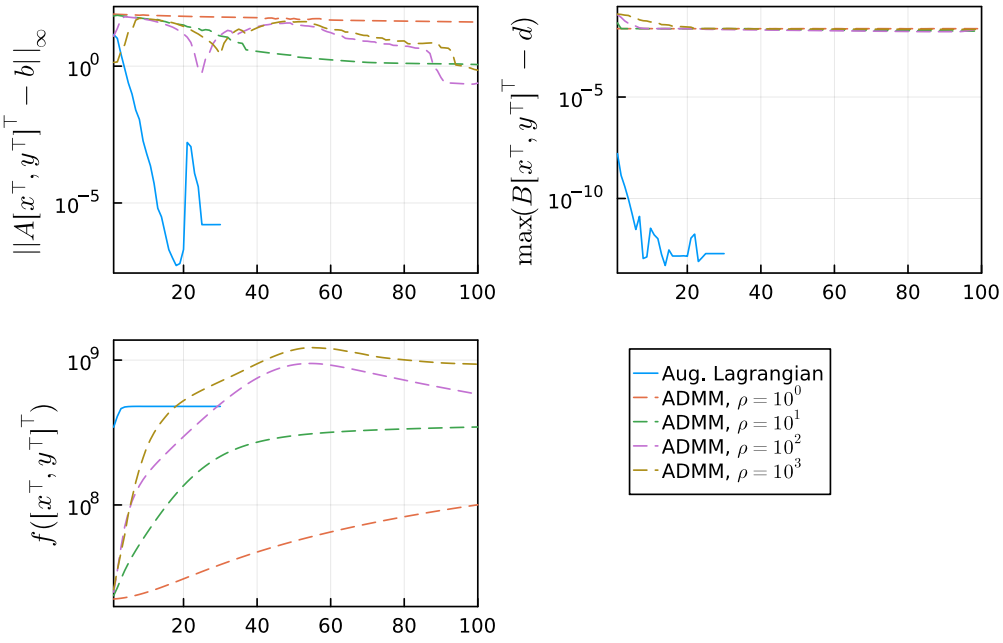

Figure 3: Convergence for $|\mathcal{S}| = 180$ buildings.

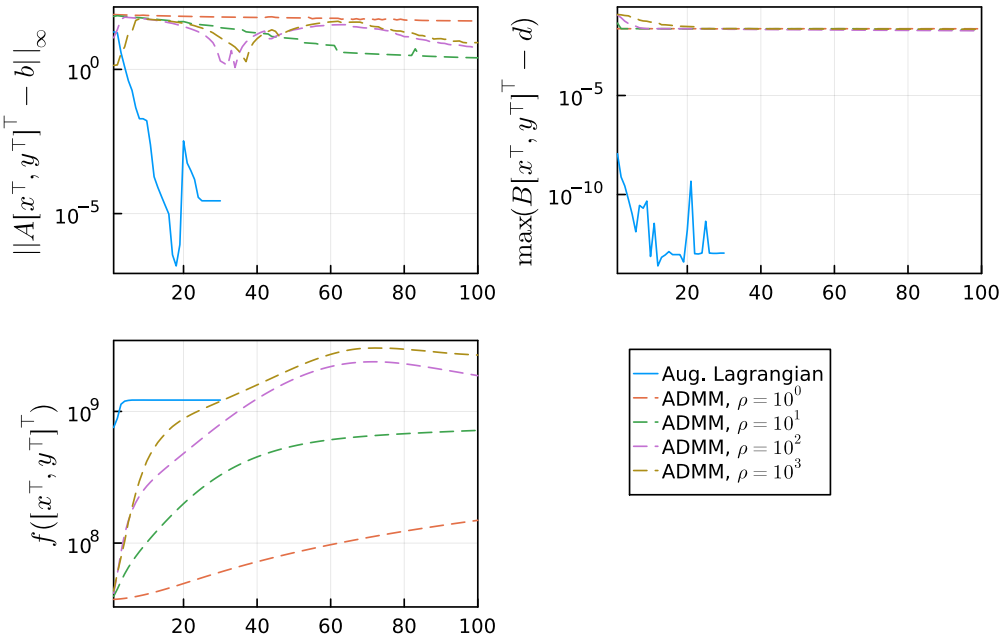

Figure 4: Convergence for $|\mathcal{S}| = 300$ buildings.

Figure 2 illustrates the numerical performance of both primal decomposition variants and ADMM. Figures 3-4 show the AL formulation only, since the $\ell 1$ formulation runs out of memory for these cases. The constraint violations for the equality constraints (1b), $\left\|[A^x \;\; A^y][x^\top \;\; y^\top]^\top - b\right\|_\infty$, for the inequality constraints (1c), $\max([B^x \;\; B^y][x^\top \;\; y^\top]^\top - b)$, and the value of the cost function $f([x^\top, y^\top]^\top)$ from (1a) are displayed, where the x-axis shows the iteration count. One can observe that the primal decomposition schemes achieve a high degree of feasibility in less than 10 iterations for all cases. Moreover, the optimality gap $(f([x^\top, y^\top]^\top) - f([x^{\star\top}, y^{\star\top}]^\top))/f([x^{\star\top}, y^{\star\top}]^\top)$ is below 0.01% in less than 10 iterations for both primal decomposition variants and for all $|\mathcal{S}|$, where $[x^{\star\top}, y^{\star\top}]^\top$ is computed via `Ipopt`. For ADMM, infeasibility and the optimality gap stay large independently of the choice of the tuning parameter ρ.

Remark 2 (Scaling of the $\ell 1$ formulation)**.** *The reason for the poor scaling of the $\ell 1$-formulation is two-fold: First, the relaxation* (8) *introduces $2n_y$ additional slack variables and inequality constraints. Hence, the KKT system in the subproblems defined via* (26) *has a larger size than the KKT system we get with the AL formulation* (23)*. Moreover, the additional inequality constraints potentially lead to smaller stepsizes due to the fraction-to-boundary rule [29, Eq 19.9]. Hence, more iterations in the subproblems are required compared to the AL formulation.*

Discussion of Algorithmic Properties

Next, we discuss algorithm properties in view of the desirable properties from Section 1.

Computation Times

Comparing computation times between algorithms is difficult, since the numerical performance strongly depends on the implementation. Nonetheless, we would like to provide some timing information to underline the potential of primal decomposition methods. Table 2 shows the computation times and the number of iterations for all algorithms. `Ipopt` is terminated at optimality with default settings and the computation time for the primal decomposition schemes are evaluated once an optimality gap of 10^{-4} and a maximum constraint violation for equality/inequality constraints of 10^{-5} is reached. ADMM is terminated after a fixed amount of 100 iterations. One can observe that the AL formulation and `Ipopt` with parallel linear algebra have similar computation times independently of $|\mathcal{S}|$. Although fast computation is not our primary focus here, this indicates the potential of primal decomposition for large-scale optimization.

`Ipopt` requires more than 100 iterations in all cases, which is unusual for second-order interior-point methods. Both ADMM and the open-source solver `OSQP` are not able to solve these problems to a sufficient accuracy with a reasonable number of iterations (4,000 for `OSQP`). This indicates that these problems are rather difficult, which might be due to the large number of inequality constraints coming from the temperature bounds. On the other hand, the primal decomposition schemes require far less iterations. A possible explanation for the behavior of `Ipopt` is that interior-point methods typically

Table 3: Internal timing (%) for the AL formulation, 30 cores.

$\lvert\mathcal{S}\rvert$	sensitivity eval.	local sol.	coord.	line search	other
300	68.10	6.66	6.10	17.84	1.30
180	41.58	19.86	9.02	26.89	2.65
30	4.37	6.69	9.35	79.29	0.30

choose the smallest stepsize such that no inequality constraint is violated (fraction-to-boundary rule) [29, Eq 19.9]. This can lead to a slow progress since in this case only small steps are taken. In primal decomposition schemes, each subproblem chooses "its own" stepsize independently which might be one explanation for faster progress.

Internal timings for the AL formulation and different sizes $|\mathcal{S}|$ are shown in Table 3. Here, the time spent in the coordination problem (13) and in the local solvers stays relatively constant for varying $|\mathcal{S}|$. The time spent for sensitivity computation, however, increases significantly with $|\mathcal{S}|$. One explanation for that is that the complexity in the backsolves for computing the Hessian via (12) increases with $O(n_y^3)$. In case the backsolves can be parallelized, for example if the subproblems themselves use multi-threading in a cluster environment, computation time can be reduced.

Communication Requirements

Both primal decomposition schemes require the communication of the gradient and the Hessian of the optimal value functions in each outer iteration. In the line-search, only the current iterate has to be communicated to the subsystems and the resulting optimal value of the local problem is communicated back. Hence, the required amount of communication in one iteration is higher than in ADMM, where only the current iterates of the coupling variable y has to be communicated. However, the difference in total communication—i.e. the number of bits to be communicated until a certain solution accuracy is reached—is still often smaller since the primal decomposition schemes require a significantly smaller number of iterations for a given target solution quality.

Feasibility and Optimality

Reaching feasibility fast is often crucial in the context of infrastructure systems to ensure system stability. Here, primal decomposition can shine. For all case studies, one can observe a high degree of primal feasibility in 10-20 iterations. ADMM requires far more iterations to reach a sufficient degree of feasibility, which is a known limitation of ADMM [4].

8. Conclusion and Outlook

We have presented two primal decomposition schemes to solve large-scale QPs for the operation of infrastructure networks. The developed methods are proven to converge

globally to the optimal solution. Numerical experiments have demonstrated their potential for solving large-scale QPs in a small number of iterations to a high degree of feasibility and optimality, which distinguishes them from classical distributed methods such as ADMM. Moreover, we have shown that primal decomposition based on augmented Lagrangians has numerical benefits compared to the classical $\ell 1$-formulation.

Future work will further improve implementation aspects of the developed primal decompositions schemes. Here, sparse backsolves or quasi-Newton Hessian approximations have the potential to greatly reduce the computation time spent in Hessian computation.

A. Sensitivities and Precomputation for Augmented Lagrangians

Observe that for computing $\nabla_y \Phi_i^\delta$ and $\nabla_{yy}^2 \Phi_i^\delta$ in (10) and (11), the partial derivatives of the implicit function F_i^δ and L_i are required. Next, we derive these quantities for the two relaxed local problems (6) and (8).

For (6), the Lagrangian (omitting arguments) reads

$$\begin{aligned} L_i^{\delta,\rho} \doteq & \frac{1}{2}\begin{bmatrix} x_i \\ y \\ z_i \end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} & 0 \\ H_i^{xy\top} & H_i^{yy}+\rho I & -\rho I \\ 0 & -\rho I & \rho I \end{bmatrix} \begin{bmatrix} x_i \\ y \\ z_i \end{bmatrix} + \begin{bmatrix} h_i^x \\ h_i^y + \lambda_i^k \\ -\lambda_i^k \end{bmatrix}^\top \begin{bmatrix} x_i \\ y \\ z_i \end{bmatrix} \\ & - \delta \mathbb{1}^\top \ln(s_i) + \gamma_i^\top \left(\begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - b_i \right) \\ & + \mu_i^\top (\begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top + s_i - d_i). \end{aligned}$$

Hence, the local KKT conditions read

$$T_i^{\delta,\rho}(q_i^\star, y) \doteq \begin{bmatrix} H_i^{xx} x_i^\star + H_i^{xy} y + h_i^x + A_i^{x\top}\gamma_i^\star + B_i^{x\top}\mu_i^\star \\ \rho(z_i^\star - y) - \lambda_i^k + A_i^{y\top}\gamma_i^\star + B_i^{y\top}\mu_i^\star \\ -(S_i^\star)^{-1}\delta\mathbb{1} + \mu_i^\star \\ \begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^{\star\top} & z_i^{\star\top} \end{bmatrix}^\top - b_i \\ \begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^{\star\top} & z_i^{\star\top} \end{bmatrix}^\top + s_i^\star - d_i \end{bmatrix} = 0,$$

where $q_i^\top \doteq \begin{bmatrix} x_i^\top, z_i^\top, s_i^\top, \gamma_i^\top, \mu_i^\top \end{bmatrix}$. Moreover,

$$\nabla_{q_i} T_i^{\delta,\rho}(q_i, y) = \begin{bmatrix} H_i^{xx} & 0 & 0 & A_i^{x\top} & B_i^{x\top} \\ 0 & \rho I & 0 & A_i^{y\top} & B_i^{y\top} \\ 0 & 0 & S_i^{-1}M_i & 0 & I \\ A_i^x & A_i^y & 0 & 0 & 0 \\ B_i^x & B_i^y & I & 0 & 0 \end{bmatrix}, \tag{19}$$

$$\nabla_y T_i^{\delta,\rho}(q_i, y) = \begin{bmatrix} H_i^{xy\top} & -\rho I & 0 & 0 & 0 \end{bmatrix}^\top, \tag{20}$$

where $M_i = \operatorname{diag}(\mu_i)$. Moreover, by (10),

$$\nabla_y \Phi_i^{\delta,\rho}(y) = \nabla_y L_i{}^{\delta,\rho}(q_i^\star(y); y) = (H_i^{yy} + \rho I)y + H_i^{xy} x_i^\star + h_i^y + \lambda_i^k - \rho z_i^\star.$$

Furthermore, by (11),

$$\nabla^2_{yy}\Phi_i^{\delta,\rho}(y) = H_i^{yy} + \rho I + [H_i^{xy\top} \; -\rho I \;\; 0 \;\; 0 \;\; 0]\, \nabla_y q_i^\star(y), \tag{21}$$

where $q_i^\star(y)$ can be computed by the system of linear equations

$$\nabla_{q_i} T_i^{\delta,\rho}(q_i^\star, y)\, \nabla_y q_i^\star(y) = -\nabla_y T_i^{\delta,\rho}(q_i^\star, y) \tag{22}$$

Precomputation for Hessian Evaluation

Next, we show how to precompute matrices to make (22) easier to solve, cf. [35, Sec IV]. We assume that H_i^{xx} is invertible—if this is not the case, one can use the variant without precomputation. Recall that by (21), we need to compute $H_i^{xy\top}\nabla_y x_i^\star - \rho \nabla_y z_i^\star$, where $(\nabla_y x_i^\star, \nabla_y z_i^\star)$ are given by (22):

$$\begin{bmatrix} H_i^{xx} & 0 & 0 & A_i^{x\top} & B_i^{x\top} \\ 0 & \rho I & 0 & A_i^{y\top} & B_i^{y\top} \\ 0 & 0 & S_i^{\star -1} M_i^\star & 0 & I \\ A_i^x & A_i^y & 0 & 0 & 0 \\ B_i^x & B_i^y & I & 0 & 0 \end{bmatrix} \begin{bmatrix} \nabla_y x_i^\star \\ \nabla_y z_i^\star \\ \nabla_y s_i^\star \\ \nabla_y \gamma_i^\star \\ \nabla_y \mu_i^\star \end{bmatrix} = \begin{bmatrix} -H_i^{xy} \\ \rho I \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{23}$$

By the third block-row, we have that $\nabla_y s_i^\star = -M_i^{\star -1} S_i^\star \nabla_y \mu_i^\star$. This yields

$$\begin{bmatrix} H_i^{xx} & 0 & A_i^{x\top} & B_i^{x\top} \\ 0 & \rho I & A_i^{y\top} & B_i^{y\top} \\ A_i^x & A_i^y & 0 & 0 \\ B_i^x & B_i^y & 0 & -M_i^{\star -1} S_i^\star \end{bmatrix} \begin{bmatrix} \nabla_y x_i^\star \\ \nabla_y z_i^\star \\ \nabla_y \gamma_i^\star \\ \nabla_y \mu_i^\star \end{bmatrix} = \begin{bmatrix} -H_i^{xy} \\ \rho I \\ 0 \\ 0 \end{bmatrix}.$$

Since H_i^{xx} is invertible, we have

$$\begin{bmatrix} \nabla_y x_i^\star \\ \nabla_y z_i^\star \end{bmatrix} = \underbrace{\begin{bmatrix} H_i^{xx} & 0 \\ 0 & \rho I \end{bmatrix}^{-1}}_{\doteq P_i^{-1}} \left(\begin{bmatrix} -H_i^{xy} \\ \rho I \end{bmatrix} - \underbrace{\begin{bmatrix} A_i^x & A_i^y \\ B_i^x & B_i^y \end{bmatrix}^\top}_{\doteq K_i^\top} \begin{bmatrix} \nabla_y \gamma_i^\star \\ \nabla_y \mu_i^\star \end{bmatrix} \right). \tag{24}$$

Employing the Schur complement with respect to the first two block rows yields

$$\left[K_i P_i^{-1} K_i^\top + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & M_i^{\star -1} S_i^\star \end{bmatrix}}_{\doteq W_i} \right] \begin{bmatrix} \nabla_y \gamma_i^\star \\ \nabla_y \mu_i^\star \end{bmatrix} = \underbrace{\begin{bmatrix} -A_i^x H_i^{xx-1} H_i^{xy} + A_i^y \\ B_i^x H_i^{xx-1} H_i^{xy} + B_i^y \end{bmatrix}}_{\doteq R_i}.$$

Observe that H_i^{xx-1}, K_i and R_i can be precomputed. Moreover, the above system of linear equations has significantly less decision variables compared to (22) and is in addition positive definite under Assumption 1. This allows to use the Cholesky or Bunch-Kaufmann (LDL) factorization instead of LU.

Precomputation for the Local Solvers

Similar to the above, we can use precomputation for speeding up the solution of the local optimization problems (6) in case interior-point methods are used. Here, we need to compute Newton steps $\nabla_{q_i} T_i^{\delta,\rho}(q_i, y)\,\Delta q_i = -T_i^{\delta,\rho}(q_i, y)$, i.e.

$$\begin{bmatrix} H_i^{xx} & 0 & A_i^{x\top} & B_i^{x\top} \\ 0 & \rho I & A_i^{y\top} & B_i^{y\top} \\ A_i^x & A_i^y & 0 & 0 \\ B_i^x & B_i^y & 0 & -M_i^{-1}S_i \end{bmatrix} \begin{bmatrix} \Delta x_i \\ \Delta z_i \\ \Delta\gamma_i \\ \Delta\mu_i \end{bmatrix} = \tag{25}$$
$$-\begin{bmatrix} H_i^{xx}x_i + H_i^{xy}y + h_i^x + A_i^{x\top}\gamma_i + B_i^{x\top}\mu_i \\ \rho(z_i - y) - \lambda_i^k + A_i^{y\top}\gamma_i + B_i^{y\top}\mu_i \\ \begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - b_i \\ \begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top + s_i - d_i + M_i^{-1}(\delta\mathbb{1} - S_i\mu_i) \end{bmatrix} \doteq \begin{bmatrix} g_1 \\ g_2 \\ h_1 \\ h_2 \end{bmatrix},$$

where we have eliminate the third block-row via $\Delta s_i = -M_i^{-1}(S_i\Delta\mu_i - \delta\mathbb{1} + S_i\mu_i)$. Solving for the first two block-rows yields

$$\begin{bmatrix} \Delta x_i \\ \Delta z_i \end{bmatrix} = P_i^{-1}\left(\begin{bmatrix} g_1 \\ g_2 \end{bmatrix} - K_i^\top \begin{bmatrix} \Delta\gamma_i \\ \Delta\mu_i \end{bmatrix}\right),$$

where P_i^{-1} and K_i are from (24). Again, employing the Schur-complement with respect to the first two block rows yields

$$\left(K_i P_i^{-1} K_i^\top + W_i\right)\begin{bmatrix} \Delta\gamma_i \\ \Delta\mu_i \end{bmatrix} = K_i P_i^{-1} \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} - \begin{bmatrix} h_1 \\ h_2 \end{bmatrix}.$$

Observe that the above is again a small system of linear equations with positive definite coefficient matrix, which allows for using Cholesky factorization and precomputed H_i^{xx-1} and K_i.

Sensitivities for the $\ell 1$ Formulation

The Lagrangian to (8) reads

$$\begin{aligned} L_i = {} & \frac{1}{2}\begin{bmatrix} x_i \\ z_i \end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} \\ H_i^{xy\top} & H_i^{yy} \end{bmatrix} \begin{bmatrix} x_i \\ z_i \end{bmatrix} + \begin{bmatrix} h_i^x \\ h_i^y \end{bmatrix}^\top \begin{bmatrix} x_i \\ z_i \end{bmatrix} - \delta(\mathbb{1}^\top \ln(s_i) + \mathbb{1}^\top \ln(v_i) + \mathbb{1}^\top \ln(w_i)) \\ & + \gamma_i^\top\left(\begin{bmatrix} A_i^x & A_i^y \end{bmatrix}\begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - b_i\right) + \mu_i^\top(\begin{bmatrix} B_i^x & B_i^y \end{bmatrix}\begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top + s_i - d_i) \\ & + \bar{\lambda}\mathbb{1}^\top(v_i + w_i) + \chi_i^\top(y - z_i - v_i + w_i). \end{aligned}$$

Hence, the KKT conditions require

$$T_i^{\delta,\bar{\lambda}}(u_i^\star, y) \doteq \begin{bmatrix} H_i^{xx} x_i^\star + H_i^{xy} z_i^\star + h_i^x + A_i^{x\top}\gamma_i^\star + B_i^{x\top}\mu_i^\star \\ H_i^{yy} z_i^\star + H_i^{xy\top} x_i^\star + h_i^y - \chi_i^\star + A_i^{y\top}\gamma_i^\star + B_i^{y\top}\mu_i^\star \\ -S_i^{-1}\delta\mathbb{1} + \mu_i^\star \\ -\delta V_i^{-1}\mathbb{1} + (\bar{\lambda}\mathbb{1} - \chi_i^\star) \\ -\delta W_i^{-1}\mathbb{1} + (\bar{\lambda}\mathbb{1} + \chi_i^\star) \\ \begin{bmatrix} A_i^x & A_i^y \end{bmatrix}\begin{bmatrix} x_i^{\star\top} & z_i^{\star\top} \end{bmatrix}^\top - b_i \\ \begin{bmatrix} B_i^x & B_i^y \end{bmatrix}\begin{bmatrix} x_i^{\star\top} & z_i^{\star\top} \end{bmatrix}^\top + s_i^\star - d_i \\ y - z_i^\star - v_i^\star + w_i^\star \end{bmatrix} \stackrel{!}{=} 0,$$

where $u_i^\top \doteq \begin{bmatrix} x_i^\top, z_i^\top, s_i^\top, v_i^\top, w_i^\top, \gamma_i^\top, \mu_i^\top, \chi_i^\top \end{bmatrix}$. Thus,

$$\nabla_{u_i} T_i^{\delta,\bar{\lambda}}(u_i, y) = \begin{bmatrix} H_i^{xx} & H_i^{xy} & 0 & 0 & 0 & A_i^{x\top} & B_i^{x\top} & 0 \\ H_i^{xy\top} & H_i^{yy} & 0 & 0 & 0 & A_i^{y\top} & B_i^{y\top} & -I \\ 0 & 0 & S_i^{-1}M_i & 0 & 0 & 0 & I & 0 \\ 0 & 0 & 0 & V_i^{-1}(\bar{\lambda}I - X_i) & 0 & 0 & 0 & -I \\ 0 & 0 & 0 & 0 & W_i^{-1}(\bar{\lambda}I + X_i) & 0 & 0 & I \\ A_i^x & A_i^y & 0 & 0 & 0 & 0 & 0 & 0 \\ B_i^x & B_i^y & I & 0 & 0 & 0 & 0 & 0 \\ 0 & -I & 0 & -I & I & 0 & 0 & 0 \end{bmatrix}, \tag{26}$$

where $V_i = \operatorname{diag}(v_i)$, $W_i = \operatorname{diag}(w_i)$, and $X_i = \operatorname{diag}(\chi_i)$. Moreover,

$$\nabla_y T_i^{\delta,\bar{\lambda}}(u_i, y) = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; I]^\top. \tag{27}$$

Furthermore, $\nabla_y \Phi_i^{\delta,\bar{\lambda}}(y) = \nabla_y L_i = \chi_i$, $\nabla_{yy} L_i = 0$, and $\nabla_{y u_i^\star} L_i = [0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; I]$. Thus, by (11),

$$\nabla_{yy} \Phi_i^{\delta,\bar{\lambda}}(y) = \nabla_y \chi_i^\star(y). \tag{28}$$

B. Proof of Lemma 1

First, we will show that $Z^\top (ZCZ^\top)^{-1} Z = Z^\top Z C^{-1} Z^\top Z$ for a regular, symmetric $C \in \mathbb{R}^{n\times n}$, $Z \in \mathbb{R}^{m\times n}$ with $m < n$. Consider a re-ordered eigendecomposition $C = Q\Lambda Q^\top$ and partition $Q = [Q_1 \;\; Q_2]$, $\Lambda_i = \operatorname{blkdiag}(\Lambda_1, \Lambda_2)$ such that Q_2 is a nullspace-basis of Z, i.e. $ZQ_2 = 0$. Hence, we have $ZCZ^\top = Z[Q_1 \;\; Q_2] \operatorname{blkdiag}(\Lambda_1, \Lambda_2)[Q_1 \;\; Q_2]^\top Z_i^\top = ZQ_1\Lambda_1 Q_1 Z^\top$ since $ZQ_2 = 0$. Thus, $Z^\top (ZCZ^\top)^{-1} Z = Z^\top Z Q_1 \Lambda_1^{-1} Q_1^\top Z^\top Z$. Again, since $ZQ_2 = 0$, we can expand this expression to $Z^\top Z[Q_1 \;\; Q_2] \operatorname{blkdiag}(\Lambda_1^{-1}, \Lambda_2^{-1})[Q_1 \;\; Q_2]^\top Z^\top Z = Z^\top Z C^{-1} Z^\top Z$.

Proof of a): By (21), we need $\nabla_y q_i^\star(y)$ for computing $\nabla_{yy}\Phi_i^{\delta,\rho}$, where $\nabla_y q_i^\star(y)$ is defined by (22). Define $C_i \doteq \operatorname{blkdiag}(H_i^{xx}, \rho I, S_i^{-1}M_i))$, $D_i \doteq \begin{bmatrix} A_i^x & A_i^y & 0 \\ B_i^x & B_i^y & I \end{bmatrix}$, and $E_i \doteq [H_i^{xy\top} \; -\rho I \;\; 0]^\top$. Consider (20) and parametrize $(\nabla_y x_i^{\star\top}, \nabla_y z_i^{\star\top}, \nabla_y s_i^{\star\top})^\top \doteq Z_i P_i$, where

Z_i is a nullspace matrix to D_i, i.e., the columns of Z_i form an orthogonal basis of the nullspace of D_i and $P_i \in \mathbb{R}^{(n_{xi}+n_y-\mathrm{nr}(A_i^x))\times n_y}$ is an auxiliary matrix. Using the above parametrization in (22) and multiplying with $Z_i^\top$ yields $Z_i^\top C_i Z_i P_i = -Z_i^\top E_i$ by $Z_i^\top D_i^\top = 0$. Since $s_i, \mu_i > 0$ and Assumption 1 holds, we have $C_i \succ 0$ and thus $Z_i^\top C_i Z_i$ is invertible by full rank of Z_i. Hence, by (21) and the above derivation, $\nabla_{yy}\Phi_i^{\delta,\rho}(y) = H_i^{yy} + \rho I - E_i^\top Z_i (Z_i^\top C_i Z_i)^{-1} Z_i^\top E_i = H_i^{yy} + \rho I - E_i^\top Z_i Z_i^\top C_i^{-1} Z_i Z_i^\top E_i$. Notice that $Z_i Z_i^\top$ is a diagonal matrix with $\mathrm{rank}(Z_i)$ ones and $\dim(C_i) - \mathrm{rank}(Z_i)$ zeros. Hence, since C_i is positive definite, it suffices to show that $\nabla_{yy}\Phi_i^{\delta,\rho}(y) \succ 0$ for the worst case, i.e. $Z_i Z_i^\top = I$ (no constraints). In this case we have $\nabla_{yy}\Phi_i^{\delta,\rho}(y) = H_i^{yy} - H_i^{xy\top}(H_i^{xx})^{-1} H_i^{xy} \succ 0$ by the definition of E_i, C_i, by Assumption 1 a) and the Schur-complement Lemma [36, A.14].

Proof of b): By (28), we need to show that $\nabla_y \chi_i^\star(y) \succ 0$, which can be computed by the system of linear equations (26), (27). Define $F_i = \mathrm{blkdiag}\left(\begin{bmatrix} H_i^{xx} & H_i^{xy} \\ H_i^{xy\top} & H_i^{yy} \end{bmatrix}, S_i^{-1} M_i, V_i^{-1}(\bar\lambda - X_i), W_i^{-1}(\bar\lambda + X_i) \right)$ and $G_i \doteq \begin{bmatrix} A_i^x & A_i^y & 0 & 0 & 0 \\ B_i^x & B_i^y & I & 0 & 0 \\ 0 & -I & 0 & -I & I \end{bmatrix}$. By Assumption 1, $s_i, v_i, w_i, \mu_i > 0$, and $\bar\lambda > \max_j |[\chi_i]_j|$, we have that $F_i \succ 0$. Hence, $\left(\nabla_y x_i^{\star\top}, \nabla_y z_i^{\star\top}, \nabla_y s_i^{\star\top}, \nabla_y v_i^{\star\top}, \nabla_y w_i^{\star\top}\right)^\top = -F_i^{-1} G_i (\nabla_y \gamma_i^{\star\top}, \nabla_y \mu_i^{\star\top}, \nabla_y \chi_i^{\star\top})$. Thus, $G_i^\top F_i^{-1} G_i (\nabla_y \gamma_i^{\star\top}, \nabla_y \mu_i^{\star\top}, \nabla_y \chi_i^{\star\top}) = [0 \;\; 0 \;\; I]^\top$. Since $F_i^{-1} \succ 0$ and by full rank of G_i from Assumption 1, $G_i^\top F_i^{-1} G_i \succ 0$ and thus $(\nabla_y \gamma_i^{\star\top}, \nabla_y \mu_i^{\star\top}, \nabla_y \chi_i^{\star\top}) = (G_i^\top F_i^{-1} G_i)^{-1} [0 \;\; 0 \;\; I]^\top$. Since $(G_i^\top F_i^{-1} G_i)^{-1} \succ 0$, all leading principle minors of this matrix must be positive definite by Sylvester's criterion [37, Col 7.1.5]. By variable reordering, the assertion follows.

C. Solution of (1) via ADMM

We derive a distributed ADMM version for (1) as a baseline for numerical comparison. Consider (1), introduce auxiliary variables $z_i \in \mathbb{R}^{n_y}$ and consensus constraints $y = z_1 = \cdots = z_S \mid \lambda_1, \ldots, \lambda_S$. This yields

$$\min_{x,y,z} \sum_{i\in\mathcal{S}} \frac{1}{2} \begin{bmatrix} x_i \\ z_i \end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} \\ H_i^{xy\top} & H_i^{yy} \end{bmatrix} \begin{bmatrix} x_i \\ z_i \end{bmatrix} + \begin{bmatrix} h_i^x \\ h_i^y \end{bmatrix}^\top \begin{bmatrix} x_i \\ z_i \end{bmatrix} \tag{29a}$$

$$\text{subject to} \quad \begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - b_i = 0, \;\; i \in \mathcal{S}, \tag{29b}$$

$$\begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - d_i \le 0, \;\; i \in \mathcal{S}, \tag{29c}$$

$$A^y y - b^y = 0, \quad B^y y - d^y \le 0 \tag{29d}$$

$$z_i = y, \;\; i \in \mathcal{S}. \tag{29e}$$

The augmented Lagrangian with respect to $y - z_i = 0$ reads

$$L^\rho = \sum_{i\in\mathcal{S}} \phi_i^x(x_i, z_i) + \phi^y(y) + \lambda_i^{k\top}(y - z_i) + \frac{\rho}{2}\|y - z_i\|^2,$$

where ϕ_i^x are defined by (29a)-(29c) and ϕ^y is the indicator function for (29d). Minimizing L^ρ w.r.t. (x_i, z_i) for fixed (y^k, λ_i^k) yields for all $i \in \mathcal{S}$

$$\begin{aligned}
&(x_i^{k+1}, z_i^{k+1}) = \\
&\arg\min_{x_i, z_i} \frac{1}{2}\begin{bmatrix} x_i \\ z_i \end{bmatrix}^\top \begin{bmatrix} H_i^{xx} & H_i^{xy} \\ H_i^{xy\top} & H_i^{yy} + \rho I \end{bmatrix} \begin{bmatrix} x_i \\ z_i \end{bmatrix} + \begin{bmatrix} h_i^x \\ h_i^y - \lambda_i^k - \rho y^k \end{bmatrix}^\top \begin{bmatrix} x_i \\ z_i \end{bmatrix} \\
&\text{subject to} \quad \begin{bmatrix} A_i^x & A_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - b_i = 0, \\
&\qquad\qquad \begin{bmatrix} B_i^x & B_i^y \end{bmatrix} \begin{bmatrix} x_i^\top & z_i^\top \end{bmatrix}^\top - d_i \leq 0.
\end{aligned} \tag{30}$$

Minimising L^ρ w.r.t. y for fixed $(x_i^{k+1}, z_i^{k+1}, \lambda_i^k)$ yields

$$\begin{aligned}
y^{k+1} = \arg\min_y \sum_{i \in \mathcal{S}} \frac{\rho}{2}\, y^\top y + (\lambda_i^k - \rho z_i^{k+1})^\top y \\
\text{subject to} \quad A^y y - b^y = 0, \quad B^y y - d^y \leq 0.
\end{aligned} \tag{31}$$

Finally, the Lagrange multiplier update reads

$$\lambda_i^{k+1} = \lambda_i^k + \rho(y^{k+1} - z_i^{k+1}), \qquad i \in \mathcal{S}. \tag{32}$$

The update rules (30)-(32) define the ADMM iterations. Note that (30) and (32) can be executed locally for all $i \in \mathcal{S}$, whereas (31) defines the global coordination step.

References

[1] F. Capitanescu, "TSO–DSO interaction: Active distribution network power chart for TSO ancillary services provision," *Electric Power Systems Research*, vol. 163, pp. 226–230, 2018.

[2] M. Kalantar-Neyestanaki, F. Sossan, M. Bozorg, and R. Cherkaoui, "Characterizing the Reserve Provision Capability Area of Active Distribution Networks: A Linear Robust Optimization Method," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2464–2475, 2020.

[3] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Operations research*, vol. 11, no. 3, pp. 399–417, 1963.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[5] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall Englewood Cliffs, NJ, 1989, vol. 23.

[6] T. Erseghe, "Distributed Optimal Power Flow Using ADMM," *IEEE Transactions on Power Systems*, vol. 29, no. 5, pp. 2370–2380, 2014.

[7] B. Kim and R. Baldick, "A comparison of distributed optimal power flow algorithms," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 599–604, 2000.

[8] S. Shin, C. Coffrin, K. Sundar, and V. M. Zavala, "Graph-Based Modeling and Decomposition of Energy Infrastructures," *IFAC-PapersOnLine*, 16th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2021, vol. 54, no. 3, pp. 693–698, 2021.

[9] J. Huang, Z. Li, and Q. H. Wu, "Coordinated dispatch of electric power and district heating networks: A decentralized solution using optimality condition decomposition," *Applied Energy*, vol. 206, pp. 1508–1522, 2017.

[10] Y. Cao, W. Wei, L. Wu, S. Mei, M. Shahidehpour, and Z. Li, "Decentralized Operation of Interdependent Power Distribution Network and District Heating Network: A Market-Driven Approach," *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5374–5385, 2019.

[11] B. Coulbeck, M. Brdyś, C. H. Orr, and J. P. Rance, "A hierarchical approach to optimized control of water distribution systems: Part I decomposition," *Optimal Control Applications and Methods*, vol. 9, no. 1, pp. 51–61, 1988.

[12] N. Chiang, C. G. Petra, and V. M. Zavala, "Structured nonconvex optimization of large-scale energy systems using PIPS-NLP," in *2014 Power Systems Computation Conference*, 2014, pp. 1–7.

[13] V. M. Zavala, C. D. Laird, and L. T. Biegler, "Interior-point decomposition approaches for parallel solution of large-scale nonlinear parameter estimation problems," *Chemical Engineering Science*, vol. 63, no. 19, pp. 4834–4845, 2008.

[14] D. K. Varvarezos, L. T. Biegler, and I. E. Grossmann, "Multiperiod design optimization with SQP decomposition," *Computers & Chemical Engineering*, An International Journal of Computer Applications in Chemical Engineering, vol. 18, no. 7, pp. 579–595, 1994.

[15] A. Engelmann, Y. Jiang, T. Mühlpfordt, B. Houska, and T. Faulwasser, "Toward Distributed OPF Using ALADIN," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 584–594, 2019.

[16] B. Houska, J. Frasch, and M. Diehl, "An Augmented Lagrangian Based Algorithm for Distributed NonConvex Optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1101–1127, 2016.

[17] A. Engelmann, Y. Jiang, B. Houska, and T. Faulwasser, "Decomposition of Nonconvex Optimization via Bi-Level Distributed ALADIN," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1848–1858, 2020.

[18] A. Engelmann, G. Stomberg, and T. Faulwasser, "An essentially decentralized interior point method for control," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 2414–2420.

[19] G. Stomberg, A. Engelmann, and T. Faulwasser, "Decentralized non-convex optimization via bi-level SQP and ADMM," *IEEE 61st Annual Conference on Decision and Control (CDC)*, 2022. arXiv: 2204.08786.

[20] A. M. Geoffrion, "Primal Resource-Directive Approaches for Optimizing Nonlinear Decomposable Systems," *Operations Research*, vol. 18, no. 3, pp. 375–403, 1970.

[21] V. DeMiguel and W. Murray, "A local convergence analysis of bilevel decomposition algorithms," *Optimization and Engineering*, vol. 7, no. 2, pp. 99–133, 2006.

[22] V. DeMiguel and F. J. Nogales, "On Decomposition Methods for a Class of Partially Separable Nonlinear Programs," *Mathematics of Operations Research*, vol. 33, no. 1, pp. 119–139, 2008.

[23] V. M. Zavala, C. D. Laird, and L. T. Biegler, "A fast moving horizon estimation algorithm based on nonlinear programming sensitivity," *Journal of Process Control*, Selected Papers From Two Joint Conferences: 8th International Symposium on Dynamics and Control of Process Systems and the 10th Conference Applications in Biotechnology, vol. 18, no. 9, pp. 876–884, 2008.

[24] N. Yoshio and L. T. Biegler, "A Nested Schur decomposition approach for multiperiod optimization of chemical processes," *Computers & Chemical Engineering*, vol. 155, p. 107 509, 2021.

[25] S. Tu, A. Wächter, and E. Wei, "A Two-Stage Decomposition Approach for AC Optimal Power Flow," *IEEE Transactions on Power Systems*, vol. 36, no. 1, pp. 303–312, 2021.

[26] C. G. Petra and I. Aravena, *Solving realistic security-constrained optimal power flow problems*, 2021. arXiv: 2110.01669.

[27] F. E. Curtis, D. K. Molzahn, S. Tu, A. Wächter, E. Wei, and E. Wong, *A Decomposition Algorithm for Large-Scale Security-Constrained AC Optimal Power Flow*, 2021. arXiv: 2110.01737.

[28] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, Belmont, 1999.

[29] J. Nocedal and S. Wright, *Numerical Optimization*. Springer Science & Business Media, New York, 2006.

[30] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 1982.

[31] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[32] R. H. Byrd, G. Liu, and J. Nocedal, *On the Local Behavior of an Interior Point Method for Nonlinear Programming*. 1998.

[33] J. B. Rawlings, N. R. Patel, M. J. Risbeck, C. T. Maravelias, M. J. Wenzel, and R. D. Turney, "Economic MPC and real-time decision making with application to large-scale HVAC energy systems," *Computers & Chemical Engineering*, vol. 114, pp. 89–98, 2018.

[34] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, 2020.

[35] F. Pacaud, D. A. Maldonado, S. Shin, M. Schanen, and M. Anitescu, *A Feasible Reduced Space Method for Real-Time Optimal Power Flow*, 2021. arXiv: 2110.02590.

[36] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[37] R. A. Horn and C. R. Johnson, *Matrix Analysis*, 2nd edition. Cambridge University Press, 2013.