

Computational Performance Study on the Alternating Direction Method of Multipliers Algorithm for a Demand Response Peak Shaving Application

Sebastian Schwarz , *Member, IEEE*, and Antonello Monti , *Senior Member, IEEE*

Abstract—This article quantifies and benchmarks the computational performance of the distributed alternating direction method of multipliers (ADMM) optimization algorithm applied to a slow-dynamics demand response peak shaving application. We propose a hierarchical demand response architecture, in which a commercial aggregator acts as the supplier and system-level flexibility service provider to a portfolio of residential prosumers. The prosumers are equipped with flexible distributed energy resources that we model based on convex operation constraints. The optimal day-ahead peak shaving control of the portfolio is realized in a distributed way through an entirely parallel implementation of the distributed ADMM optimization algorithm. In this scenario, we evaluate the ADMM algorithm's overall speed of convergence and conduct multiple large-scale numerical simulation experiments on a compute cluster to compare and assess the algorithmic performance of ADMM against a centralized reference benchmark optimization. Our simulation results show that the distributed ADMM algorithm is superior compared to a purely centralized optimization. The findings are of particular use to commercial demand response aggregators that show interest in the deployment and improvement of distributed coordination and optimal control concepts for future slow-dynamics demand response applications and services.

Index Terms—Aggregator, alternating direction method of multipliers, day-ahead scheduling, demand response, distributed optimization, peak shaving, prosumer.

I. INTRODUCTION

IN THIS article, we consider slow-dynamics demand response (DR) applications, in which residential prosumers offer operational flexibility to a commercial DR aggregator in the day-ahead or intraday timeframe based on supply and flexibility purchase agreements. The DR aggregator acts as the flexibility service provider that bundles the energy generation and consumption flexibility offered by its local customers [1]. For instance, a virtual power plant (VPP) operator that provides both energy balancing and grid services to other stakeholders inside the energy system can take on the role of the DR aggregator

Manuscript received 29 April 2022; revised 11 August 2022 and 12 November 2022; accepted 28 December 2022. Date of publication 23 January 2023; date of current version 8 June 2023. This work was supported by the European Union's Horizon 2020 research and innovation programme under Grant 883710 (edgeFLEX). (*Corresponding author: Sebastian Schwarz.*)

Sebastian Schwarz is with the Institute for Automation of Complex Power Systems, E.ON Energy Research Center, RWTH Aachen University, 52074 Aachen, Germany (e-mail: sebastian.schwarz@eonerc.rwth-aachen.de).

Antonello Monti is with the Institute for Automation of Complex Power Systems, E.ON Energy Research Center, RWTH Aachen University, 52074 Aachen, Germany, and also with the Fraunhofer FIT, 52074 Aachen, Germany (e-mail: amonti@eonerc.rwth-aachen.de, antonello.monti@fit.fraunhofer.de).

Digital Object Identifier 10.1109/JSYST.2023.3234709

[2]. In this way, even small-scale distributed energy resources (DERs) can contribute to the wholesale energy market or support the provision of grid services [1], [2].

In this context, optimal dispatch and portfolio balancing applications pose a challenge, because they usually integrate a large number of residential DERs. From a mathematical optimization point of view, this circumstance in turn leads to a large number of variables as well as equality and inequality constraints that the corresponding optimization models contain. Commercial DR aggregators, thus, show an increased interest in distributed optimization methods.

In distributed optimization, a global optimization problem is partitioned into smaller subproblems that are computed independently using an iterative algorithm. The overall advantages of distributed over centralized optimization methods, as outlined in the review work in [3], include 1) the safeguard of end-consumer data, measurement information, constraints, and objective functions, 2) the improvement of robustness, i.e., single point of failure concerns, and 3) the ability to perform parallel, decomposed, and computationally distributed computations. Especially the third aspect is important for the solution of large-scale optimization problems, which usually cannot be accomplished with the pure computing capabilities of off-the-shelf mathematical optimization solvers.

The optimal balancing use case presented in [4] from the European H2020 research project *edgeFLEX* [5] is an example showing that distributed optimization is a feasible approach to tackle large-scale optimization problems. The authors' objective in [4] is to balance VPP portfolios in Germany consisting of up to 200 DERs. Because of the complexity and large size of the optimization problem defined in [4], in some cases a straightforward centralized computation provides either suboptimal or any feasible solution within a moderate real-time execution time limit. Considering that a failed optimal balancing process causes the loss of revenue, this outcome is insufficient and distributed optimization methods—albeit more difficult to realize—become the authors' preferred choices.

Motivated by the work in [4], our research work, therefore, proposes a hierarchical and distributed DR architecture that can be used as a reference to further deploy and improve distributed optimization methods and algorithms for the intelligent coordination and optimal control of DERs in power systems.

Making use of this DR architecture, the primary goal of this article is to evaluate the computational behavior of the widely recognized and accepted distributed alternating direction method of multipliers (ADMM) optimization algorithm applied to portfolio balancing applications, in particular

for large portfolios that may include hundreds of DERs. We map the ADMM algorithm to the proposed distributed DR architecture, in which a commercial DR aggregator acts as the supplier and system-level flexibility service provider to a portfolio of residential prosumers. The prosumers are equipped with flexible DERs whose operation we model using convex constraints and objective functions. The mapping of the ADMM algorithm to the proposed architecture makes it possible to formulate the optimal balancing problem as an optimal exchange problem with a separable structure. The distributed nature of ADMM thereby allows us to define the individual prosumers' optimization problems as the ADMM decomposed subproblems that we can solve independently and in parallel [6]. In this way, the application of the ADMM algorithm becomes independent of both the size of the portfolio and the local asset configurations on the prosumers' premises.

Our research specifically addresses the existing gap of the ADMM algorithm's scalability properties that have not been sufficiently well evidenced for large-scale slow-dynamics DR applications in the past. Although ADMM is known as a fast iterative decomposition algorithm that, in its standard form, comes with a near-zero optimality gap for convex optimization problems [6], there is still a lack of analyses that quantify and benchmark the ADMM's computational performance for fully parallel realizations of the algorithm. Given the proposed DR architecture, our research work evaluates the best-case algorithmic performance of ADMM for a residential DR peak shaving application with quadratic cost functions. We conduct large-scale simulation experiments on a compute cluster and formulate the trend of the dependency between optimization problem size, ADMM convergence speed, and its step size parameter. Our findings ultimately provide commercial DR aggregators with accurate algorithmic performance trends to decide on centralized versus ADMM-based approaches for real-world DR applications and services.

A. State of the Art and Advancements

Distributed optimization methods and algorithms for convex optimization problems based on Lagrangian relaxation, such as dual decomposition and ADMM, have been studied intensively in recent literature.

For example, Soares et al. [7] applied the distributed dual decomposition algorithm to deploy and activate flexibility on the residential building level from DERs. A system-level aggregator makes use of the pooled flexibility to cope with congestions in the distribution grid. Similar to our work, the approach in [7] is privacy preserving while it inherently takes into account all end-consumer constraints and objectives.

As indicated by Huang et al. [8], the dual decomposition algorithm, nevertheless, requires a strict convex optimization problem formulation and may be slow in convergence. The distributed ADMM algorithm, which we focus on in this work, overcomes these limitations by combining the effectiveness and robustness of the augmented Lagrangian approach from the method of multipliers with the distributed computational abilities of dual decomposition [3], [6].

Rivera et al. [9] used the ADMM algorithm to schedule a fleet of electric vehicles under different DR aggregator system-level goals, such as cost optimization or valley filling. Zhou et al. [10] extend the approach by also considering fast charging and electric vehicle battery degradation. In both works, the ADMM

algorithm solves an optimal power exchange problem that is similar to our work, but only considers operational flexibility deployment from electric vehicle charging. Instead, we include more diversified DERs such as electro-thermal heat pumps and photovoltaic units in the exchange process. A system-level equilibrium constraint thereby ensures the balance of power demand and supply throughout the optimization horizon.

Whereas our work focuses on residential prosumers, Faddel et al. [11] applied the ADMM algorithm to commercial buildings to optimize heating, ventilation, and air conditioning (HVAC) units. The ADMM optimization successfully reduces the commercial buildings' HVAC energy consumption, while it preserves the thermal comfort specifications of occupants on the building level as well as distribution grid constraints on the system operator level. Although the feasibility region of the standard alternating current power flow problem is nonconvex, convex relaxation approaches such as in [12] are considered frequently to make ADMM to additionally incorporate grid constraints for optimal energy management problems. In contrast to our work, however, Faddel et al. [11] and Manshadi et al. [12] do not specify a holistic architecture, which makes it rather difficult to apply and scale their proposed ADMM-based solutions.

Attarha et al. [13] further presented a robust approach based on the ADMM algorithm for the intelligent coordination of residential DERs. Using an affinely adjustable robust ADMM extension, prosumers negotiate "here-and-now" decisions in the long-term and compensate uncertainty using "wait-and-see" recourse decisions in the short-term. Although optimization under uncertainty is beyond the scope of our work, Attarha et al. [13] showed that measures to cope with uncertainty can be readily integrated into the ADMM optimization process.

We also do not consider approaches that make use of fully decentralized variants of the ADMM algorithm. Li et al. [14], for example, proposed a parallel and decentralized ADMM-based application for the optimal energy management in microgrids. In comparison to our work, such decentralized approaches do not require a central coordinator. Instead, information, such as in [14], is exchanged between neighboring terminals or agents solely. In consequence, decentralized applications typically demand for specific prerequisites on the underlying grid and communication network hierarchy such as a tree-based or radial network topology [3], [15]. However, these preconditions are usually not met in hierarchical DR architectures in which a commercial DR aggregator acts as the coordinating entity for prosumers in multiple, possibly galvanically separated, medium, and/or low voltage grids [15].

Beyond the scope of our investigations are moreover ADMM-based applications that include integer variables or nonconvex modeling approaches for DERs such as in the related works in [16], [17], and [18]. ADMM turns into a heuristic in that case where one can ensure neither convergence nor global optimality of the solution. Distributed algorithms other than ADMM, such as Dantzig-Wolfe decomposition for mixed-integer problems in [19] or the alternating direction inexact Newton (ALADIN) algorithm for nonconvex problems in [20], may in consequence constitute choices that are more suitable.

Furthermore and in view of the algorithmic performance of the ADMM algorithm, the above works [9], [10], [11], [13], [16], [17], [18] have in common that they make performance statements based on numerical simulation experiments in which the ADMM implementation still contains sequential or pseudoparallel workflows. Instead, our work presents more

absolute results based on computational performance investigations on a compute cluster of loosely-coupled machines where all ADMM subproblems run fully in parallel on dedicated Central Processing Unit (CPU) cores. We, therefore, strengthen the contributions in [9], [10], [11], [13], [16], [17], and [18] by quantifying the computational performance of the distributed ADMM algorithm and by putting it in relation to a centralized computation for larger scale residential DR peak shaving scenarios that are complex enough to be representative for portfolio sizes of up to 200 flexible prosumers. To the best of our knowledge, no comparable performance and scalability analyses of such a truly parallel ADMM application have been conducted in this research area before.

B. Contributions

From an operational point of view, this article defines and extensively discusses a scalable ADMM-based optimization algorithm for a slow-dynamics DR peak shaving application. We map the ADMM algorithm to a hierarchical and distributed DR architecture, in which residential prosumers own and operate DERs. The ADMM-based coordination strategy maintains an optimal exchange of electric power among the prosumers. The exchange process is fully privacy preserving according to the mapping of the ADMM algorithm to the distributed DR architecture.

From an algorithmic point of view, we compare the best-case computational performance of a parallel and, hence, fully distributed ADMM algorithm implementation against a centralized reference benchmark optimization and provide an accurate identification of the trend of the dependency between the ADMM speed of convergence, its step size parameter and the overall optimization problem size. We demonstrate that in certain cases the solution of large-scale peak shaving problems is only possible with the proposed ADMM algorithm.

From an architectural point of view, we propose a distributed and hierarchical reference DR architecture that supports the needs of commercial DR aggregators. The architecture makes it possible to integrate several hundred prosumers in a modular fashion independent of the specific asset configurations on the prosumers' local premises. The proposed architecture, together with our operational and algorithmic contributions, hence, facilitates the deployment and improvement of coordination and optimal control concepts for slow-dynamics DR applications.

II. PROPOSED REFERENCE DR ARCHITECTURE

For the effective deployment and bundling of residential prosumer-level flexibility, we first propose a hierarchical and distributed DR architecture based on the flexibility deployment approach provided by the European Smart Grids Task Force of the European Commission [21]. The approach in [21] defines a customer-centric energy system framework and interlinks the providers and users of operational flexibility subject to their individual roles, needs, challenges, and requirements. Inspired by that approach, our proposed DR architecture constitutes a valid reference DR architecture for the DR peak shaving application considered in this article. It is important to stress that our specific choice on the DR architecture, however, should be understood as a recommendation only, but it does not limit the generality of the presented methodology, results, and findings.

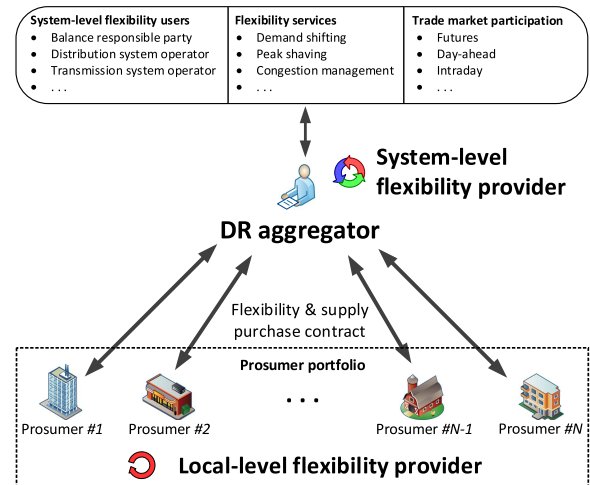


Fig. 1. Schematic of the proposed reference DR architecture based on the framework approach described in [21].

Based on [21] and as visualized in Fig. 1, a DR aggregator acts as the commercial supplier and system-level flexibility provider in the proposed DR architecture. It terminates flexibility and supply purchase contracts with its customers, i.e., a portfolio of residential prosumers as indicated by the dotted box in Fig. 1. The prosumers are the local-level flexibility providers and may be equipped with flexible (small-scale) DERs, such as photovoltaic (PV) units, electro-thermal HVAC systems, electric vehicles and/or storage units. By bundling and leveraging the operational flexibility of the local prosumers, the commercial DR aggregator—in its role of the central coordinator and according to its underlying business case—can, hence, offer flexibility and energy services to system-level flexibility users such as the balance responsible party or the distribution system operator. The services can include but are not limited to demand shifting, peak shaving or congestion management, and may also include the DR aggregator's participation in the energy trade market, compare the upper box in Fig. 1. Moreover, it is important to emphasize that, according to [21], the DR architecture is customer-centric, i.e., the privacy and sensitive data of prosumers must be protected at all times meanwhile prosumers should be empowered to actively participate in the DR solution. The latter means that prosumers must have the opportunity to pursue toward individual local-level objectives, too. For example, a prosumer local-level objective can be to minimize energy costs, reduce CO₂ emissions, or increase the self-consumption rate of the local (renewable) energy generation [21], [22]. Within the requirement of satisfying the prosumer local-level objectives, the DR aggregator hence maximizes the remaining operational flexibility of its customers to accomplish for its own objectives and services. However, each prosumer's preference between, possibly contradictory, local-level, and system-level objectives is assumed to be an integral part of the flexibility and supply purchase agreements and has been analyzed in more detail in previous work [22]. In this article, we assume that both local-level and system-level objectives are of the same preference by all prosumers, i.e., they are equally important to the prosumers.

The following section introduces the distributed ADMM algorithm formulation and links it to the proposed DR

architecture. The mapping of the ADMM algorithm to the proposed distributed DR architecture demonstrates the advantage on our architecture choice: we represent all prosumers and their local devices as mathematical sets of constraints plus objective functions, where the total number, physical types, and operational constraints of local devices become irrelevant to the DR aggregator. The proposed reference DR architecture, thus, allows commercial DR aggregators for a modular and scalable integration of hundreds of prosumers and assets while it satisfies the data privacy requirement of the prosumers. Using an approach based on a distributed optimization method such as ADMM, operational flexibility is inherently deployed at maximum level and can be used to fulfill both local-level and system-level objectives at the same time.

III. DISTRIBUTED OPTIMIZATION USING ADMM

In this section, we first define the local-level mathematical optimization model for a prosumer, and second define the system-level optimization model for the DR aggregator. The distributed DR peak shaving application under investigation, which couples the optimization models from all prosumers and the DR aggregator, is referred to as the portfolio balancing optimization problem. The primary goal of this portfolio balancing optimization is in the provision of a system-level peak shaving flexibility service to a system-level flexibility user, whilst the power demand and supply inside the DR aggregator's prosumer portfolio must be balanced based on an equilibrium constraint [23]. We reformulate this portfolio balancing optimization problem as an equivalent optimal exchange problem that has a fully separable structure. This allows us to formulate a distributed optimization approach based on the iterative exchange ADMM algorithm that satisfies the preconditions of our proposed DR architecture from Section II.

A. Definition of Variables and Domains

We define all nonbold style mathematical variables as scalars. Vice versa, all bold style mathematical variables denote time series vectors for an optimization horizon T_h that consist of T discrete time slots in total. One single time slot is defined as $t \in \{0, 1, 2, \dots, T\}$ with a fixed duration of $\Delta t = |T_h|/T$. Thus, all vectors are defined over \mathbb{R}^T , where we denote one single vector element by the superscript t . Moreover, a positive sign for power or energy represents demand/consumption, whereas a negative sign represents supply/generation.

B. Prosumer Local-Level Optimization Model

We assume that our proposed distributed DR architecture consists of one DR aggregator and N local-level prosumers. Based on the customer-centric precondition described in Section II, all N prosumers have the opportunity to perform a local-level optimization subject to their individual objectives and local constraints. For this purpose, we assume that each prosumer possesses a local-level optimization model. Similar to the work in [9], this prosumer local-level optimization model is defined as follows:

$$\min_{\mathbf{P}_{el,i}} f_i(\mathbf{P}_{el,i}) \quad (1)$$

$$\text{s.t. } \mathbf{P}_{el,i} \in \mathbb{X}_i, \quad i = 1, \dots, N. \quad (2)$$

In (1), we define the net power contribution of the i th prosumer as optimization variable $\mathbf{P}_{el,i}$, which is subject to a convex set of local prosumer constraints \mathbb{X}_i in (2). Constraint set \mathbb{X}_i includes all device constraints with which the prosumer i is equipped, i.e., \mathbb{X}_i includes the physical operation constraints for all local storage, load and/or generation assets present. In this work, we adopt state-of-the-art convex device optimization models. They are specified in the Appendix section. Moreover, function f_i in (1) represents the individual objective function of the i th prosumer. We choose a self-consumption maximization objective for the prosumer local-level optimization models based on the underlying customer-centric DR architecture. This implies that the prosumers intend to minimize their total grid feed-in per time slot, i.e., the total (renewable) excess energy generation throughout the optimization horizon T_h . We, therefore, consider the following quadratic objective function for all prosumers $i = 1, \dots, N$:

$$f_i(\mathbf{P}_{el,i}) = \sum_t (\min\{0, P_{el,i}^t\})^2. \quad (3)$$

C. DR Aggregator System-Level Optimization Model

In analogy to the prosumer local-level optimization model, the DR aggregator system-level optimization model with reference to the work in [9] is defined as follows:

$$\min_{\mathbf{P}_{el,0}} f_0(\mathbf{P}_{el,0}) \quad (4)$$

$$\text{s.t. } \mathbf{P}_{el,0} \in \mathbb{X}_0. \quad (5)$$

In (4), we define the net power contribution of the DR aggregator to the entire prosumer portfolio as optimization variable $\mathbf{P}_{el,0}$, which is subject to a convex set of DR aggregator constraints \mathbb{X}_0 in (5). Function f_0 represents the objective function of the DR aggregator, i.e., its desired flexibility or energy service to a system-level flexibility user. Since we investigate a DR peak shaving application in this article, with the goal to mitigate power consumption/generation peaks on the portfolio level, function f_0 is quadratic in our case, such that

$$f_0(\mathbf{P}_{el,0}) = \sum_t (P_{el,0}^t)^2. \quad (6)$$

Constraint set \mathbb{X}_0 , according to (5), defines the constraints on the DR aggregator's system-level power contribution $\mathbf{P}_{el,0}$, e.g., such as minimum and maximum power subscriptions. However, in the simplest case \mathbb{X}_0 defines the empty set, i.e., $\mathbb{X}_0 = \emptyset$. This means that no constraints apply to the power contribution $\mathbf{P}_{el,0}$ of the DR aggregator, which we also assume in this article.

D. Portfolio Balancing Optimization Problem

The DR aggregator's primary goal is to maximize the unused operational flexibility of its prosumer customers to provide flexibility or energy services to a system-level flexibility user. Because of the customer-centric design of our DR architecture, however, one must also account for the local objectives of the prosumers at all times. For this purpose, we adopt the definition of the customer-centric portfolio balancing optimization problem from [23], which interlinks (1)–(6) as follows:

$$\min_{\mathbf{P}_{el,0}} \min_{\mathbf{P}_{el,i}} f_0(\mathbf{P}_{el,0}) + \sum_{i=1}^N \beta_i f_i(\mathbf{P}_{el,i}) \quad (7)$$

$$\text{s.t. } \mathbf{P}_{el,0} \in \mathbb{X}_0, \mathbf{P}_{el,i} \in \mathbb{X}_i, i = 1, \dots, N \quad (8)$$

$$\text{s.t. } \mathbf{P}_{el,0}^t + \sum_{i=1}^N \mathbf{P}_{el,i}^t = 0, \forall t. \quad (9)$$

In (7), we optimize toward both the objective of the DR aggregator and the local objectives of the prosumers at the same time, where variable $\beta_i \geq 0$ defines an individual participating factor per prosumer i . Each prosumer can define its local objective function f_i that is weighted against the system-level objective by means of factor β_i . We assume that both local-level and system-level objectives are of the same importance for the prosumers in this article (see Section II), i.e., we consider $\beta_i = 1$ for all prosumers i . In (8), we integrate the convex constraint sets \mathbb{X}_i and \mathbb{X}_0 from (2) and (5), respectively. Constraint (9) represents the equilibrium constraint of the portfolio balancing problem: the DR aggregator also acts as a supplier to its prosumers, which means that it is responsible for the balance of the net power demand and supply inside the portfolio, i.e., we enforce the portfolio's residual power load to vanish [23].

The portfolio balancing optimization problem in (7)–(9) defines a centralized quadratic programming optimization problem, which one could solve using an off-the-shelf mathematical optimization solver that supports quadratic objective functions. We define this problem as our centralized reference benchmark optimization problem. Solving this problem, however, requires the commercial DR aggregator to collect all prosumer information and parameters according to variables $\mathbf{P}_{el,i}$ and sets \mathbb{X}_i , which is considered critical from the customer data privacy point of view [3]. Moreover, the portfolio balancing optimization problem becomes computationally challenging for a large number of prosumers as the problem size, in terms of optimization variables and constraints, grows proportionally with the number of prosumers considered.

E. Distributed ADMM Optimization Problem

The portfolio balancing problem in (7)–(9) is fully separable in optimization variables $\mathbf{P}_{el,0}$ and $\mathbf{P}_{el,i}$ if we rewrite it as an equivalent optimal exchange problem by introducing a new decision variable \mathbf{z}_i and the indicator function $g(\mathbf{z}_i)$ as follows:

$$\min_{\mathbf{P}_{el,0}, \mathbf{P}_{el,i}, \mathbf{z}_i} f_0(\mathbf{P}_{el,0}) + \sum_{i=1}^N \beta_i f_i(\mathbf{P}_{el,i}) + g(\mathbf{z}_i) \quad (10)$$

$$\text{s.t. } \mathbf{P}_{el,0} \in \mathbb{X}_0, \mathbf{P}_{el,i} \in \mathbb{X}_i, i = 1, \dots, N \quad (11)$$

$$\text{s.t. } \mathbf{P}_{el,i} = \mathbf{z}_i, i = 0, \dots, N \quad (12)$$

$$\text{s.t. } g(\mathbf{z}_i) = \begin{cases} 0 & \text{if } \sum_{i=0}^N \mathbf{z}_i^t = 0, \forall t \\ \infty & \text{otherwise} \end{cases} \quad (13)$$

The equivalent optimization problem in (10)–(13) takes the so-called exchange ADMM form as defined in [6] and as further extended by the works in [9], [23]. It allows us to decompose the optimization problem in (10)–(13) and to solve it in a distributed way using the distributed exchange ADMM algorithm.

It is important to mention that there are also variants of the ADMM algorithm other than exchange ADMM, which are comprehensively studied in [6]. Exchange ADMM, however,

fits well to the preconditions of our proposed DR architecture, which is why we adopt it in this article. The rationale for this is that exchange ADMM is known as a special case for optimization problems with an economic interpretation [6]. In comparison to the standard ADMM, the exchange ADMM problem formulation can be considered as the optimal exchange between goods via a price adjustment process. This means that a competitive market seeks solutions in the direction of a market equilibrium [6]. This market equilibrium is the portfolio's residual power load in our case, as defined by the modified equilibrium constraint in (12). The electric powers $\mathbf{P}_{el,0}$ and $\mathbf{P}_{el,i}$ in (11) represent the quantities that the DR aggregator and the prosumers exchange subject to the objective functions f_i in (10). In other words, the power quantities can be understood as the individual contributions of the DR aggregator and the prosumers to the global exchange process. Through the mapping of the exchange ADMM algorithm to our proposed distributed DR architecture, the DR aggregator thereby actively pursues toward this equilibrium by adjusting shared ADMM variables that are to be integrated as penalty terms in the prosumers' local objective functions f_i . The DR aggregator adjusts the shared ADMM variables up or down depending on whether there is a lack or excess of residual power on the system-level [6], but it does never control the prosumer's local devices directly.

Based on the ADMM theory provided in [6], [9], and [23], the augmented Lagrangian function for the exchange optimization problem in (10)–(13) with $f(\mathbf{x} = \mathbf{P}_{el}) = \sum_{i=0}^N f_i(\mathbf{P}_{el,i})$ is

$$\mathcal{L}_\rho(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}) = f(\mathbf{x}) + g(\mathbf{z}) + \boldsymbol{\lambda}^T (\mathbf{x} - \mathbf{z}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}\|_2^2 \quad (14)$$

where ρ is the augmented penalty parameter and $\boldsymbol{\lambda}^T$ a vector of Lagrangian variables. The iterative exchange ADMM process involves the separate minimization over the primal decision variables \mathbf{x} and \mathbf{z} from (14); followed by the maximization over $\boldsymbol{\lambda}^T$. Rivera et al. [9] further simplified this iterative process for the optimization problem in (10)–(13), which finally yields the equations of the iterative exchange ADMM algorithm, where superscript k represents the k th iteration step, as follows:

$$\begin{aligned} \mathbf{P}_{el,j}^{(k+1)} &= \underset{\mathbf{P}_{el,j}}{\operatorname{argmin}} f_j(\mathbf{P}_{el,j}) + \boldsymbol{\lambda}^{T(k)} \mathbf{P}_{el,j} \\ &\quad + \frac{1}{2} \rho \left\| \mathbf{P}_{el,j} - \left(\mathbf{P}_{el,j}^{(k)} - \bar{\mathbf{P}}_{el}^{(k)} \right) \right\|_2^2, \quad j = 0, \dots, N \end{aligned} \quad (15)$$

$$\bar{\mathbf{P}}_{el}^{(k+1)} = \frac{1}{N+1} \left[\mathbf{P}_{el,0}^{(k+1)} + \sum_{i=1}^N \left(\mathbf{P}_{el,i}^{(k+1)} \right) \right] \quad (16)$$

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \rho^{(k)} \bar{\mathbf{P}}_{el}^{(k+1)}. \quad (17)$$

Fig. 2 illustrates the mapping of the distributed exchange ADMM algorithm to the proposed distributed DR architecture including an overview on the information flow of shared ADMM variables between the prosumers and the DR aggregator. In exchange ADMM iteration step k , the DR aggregator first sends out the shared ADMM variables $\boldsymbol{\lambda}^{(k)}$ and $\bar{\mathbf{P}}_{el}^{(k)}$ to the prosumers. ADMM variable $\boldsymbol{\lambda}^{(k)}$ converges to the optimal dual variable of the problem in (10)–(13) and we can interpret it as a virtual price incentive. This virtual price incentive is the power exchange incentive in our case, i.e., the quantity that incentivizes the

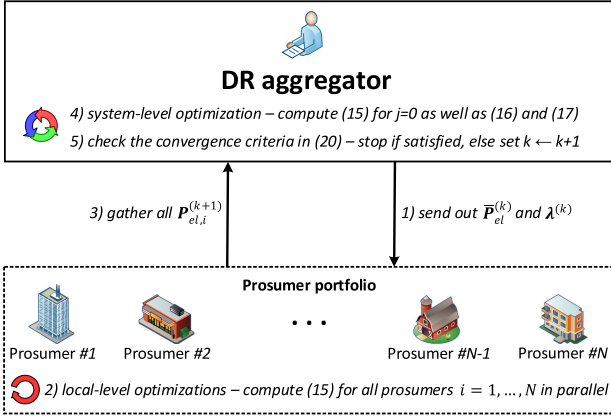


Fig. 2. Mapping of the distributed ADMM algorithm to the proposed distributed DR architecture.

prosumers to pursue toward the global power equilibrium of the portfolio based on the flexibility purchase agreements. ADMM variable $\bar{P}_{el}^{(k)}$ represents the net power residual, i.e., the current power demand versus supply imbalance inside the prosumer portfolio during iteration step k . It is desired to go to zero in order to satisfy the equilibrium constraint in (12). In a second step, all prosumers $i = 1, \dots, N$ solve their local-level optimization problems in parallel and obtain their individual optimal power contributions $P_{el,i}^{(k+1)}$ according to (15), where $j = i$. In (15), the augmented Lagrangian penalty term ρ represents the ADMM step size parameter. It is a free variable and it is sufficient to use a predefined fixed step size for the ADMM algorithm. The DR aggregator finally gathers all variables $P_{el,i}^{(k+1)}$ from its portfolio customers, performs the system-level optimization in (15) where $j = 0$, and computes the new net power residual $\bar{P}_{el}^{(k+1)}$ in (16) as well as the new virtual price incentive $\lambda^{(k+1)}$ in (17) as required for the next ADMM iteration step $k+1$ [6], [9].

For convex problems, the ADMM algorithm guarantees convergence to the optimal solution of the initial nonseparated optimization problem in (7)–(9) [3]. Therefore, it is sufficient to terminate the algorithm after a sufficiently large number of iteration steps k [3]. For this purpose, we adopt the following ADMM stopping criteria from [9] that comprise the primal residual $r^{(k)}$ in (18) and the dual residual $s^{(k)}$ in (19) as follows:

$$r^{(k)} = \left\| \bar{P}_{el}^{(k)} \right\|_2 \quad (18)$$

$$s^{(k)} = \left\| \begin{array}{c} P_{el,0}^{(k)} - P_{el,0}^{(k-1)} + \left(\bar{P}_{el}^{(k-1)} - \bar{P}_{el}^{(k)} \right) \\ \vdots \\ P_{el,N}^{(k)} - P_{el,N}^{(k-1)} + \left(\bar{P}_{el}^{(k-1)} - \bar{P}_{el}^{(k)} \right) \end{array} \right\|_2. \quad (19)$$

Based on (18), (19), the DR aggregator checks the exchange ADMM algorithm for convergence after each iteration and stops the algorithm if the following criteria are both satisfied:

$$r^{(k)} \leq \varepsilon_p, \quad \rho(N+1)s^{(k)} \leq \varepsilon_d, \quad (20)$$

where $\varepsilon_p > 0$ and $\varepsilon_d > 0$ are sufficiently small numbers and denote the user-defined ADMM tolerances on the feasibility and the optimality of the ADMM solution quality, respectively [9].

IV. ADMM COMPUTATIONAL PERFORMANCE STUDY FOR A DR PEAK SHAVING APPLICATION

In this section, we investigate the algorithmic performance of the distributed ADMM algorithm. We apply the distributed exchange ADMM algorithm defined in (15)–(20) to the equivalent optimal exchange problem in (10)–(13), where we assign every ADMM subproblem in (15), i.e., every prosumer local-level optimization problem, to a separate computing process on a compute cluster. Motivated by earlier work in [24], we consider a residential DR peak shaving application that is intended to be both a suitable and a sufficiently general test case for our analyses. This means that nonfundamental variations in the assumptions and asset configurations do not restrict and affect the generality of the presented results and findings.

A. Setup of the Residential DR Peak Shaving Application

We embed the residential DR peak shaving application into the proposed distributed DR architecture by setting the objective functions of the N prosumers and the DR aggregator as defined in (3) and (6), respectively.

We flexibly vary the number of prosumers N in the following simulation experiments to evaluate the impact of the prosumer portfolio size on the ADMM's algorithmic performance. For a fair computational comparison for different portfolio sizes N , we assume a homogeneous asset equipment over all prosumers. The assumption of homogeneity is to avoid a situation where some prosumers must compute the day-ahead schedule for a large number of local assets, while others operate only very few or even no devices. In this scenario, the overall ADMM algorithmic performance would strongly depend on the prosumer with the most complex local optimization problem formulation, which would adversely affect the average computation time per iteration, and slow down the ADMM optimization process.

In our simulation experiments, each prosumer thus represents a residential single-family house (SFH). We equip each SFH with an electro-thermal heat pump plus thermal energy storage unit, as well as with a PV unit plus stationary battery storage. The storage units provide a sufficient level of operational flexibility to the DR peak shaving application.

The required SFH data in our scenarios rests on building information and energy data provided by [25]. Based on this data as well as historical weather data for the region of Aachen, Germany for the year 2018 from [26], the thermal load demand curve per SFH is calculated. A prosumer's heat pump plus thermal energy storage unit must cover this thermal load demand during all time slots. We size the thermal energy storage unit, i.e., a standard hot water tank buffer, twice the heat pump's thermal nominal hourly energy generation. This means that a heat pump can be turned-off for at least two hours if the buffer tank is fully loaded with hot water. The power generation profile of the PV unit is calculated based on the approach provided in [27], for which we again use the data from [25] and [26]. For the stationary battery unit, we consider a state-of-the-art residential PV battery of size 13.5 kWh and 4.6 kW. For the electric load demand of a prosumer, we assume an annual electricity demand of 3 500 kWh and refer to the standard load profiles from [28].

For the sake of exemplification, we perform our DR peak shaving application in the day-ahead timeframe for one single historical day. The day-ahead optimization horizon T_h for our

TABLE I
SIZE OF THE PORTFOLIO BALANCING OPTIMIZATION MODEL

Number of prosumers N	Size of the portfolio balancing optimization model	
	Number of variables	Number of constraints
1	24 482	32 820
40	923 042	1 256 640
80	1 844 642	2 511 840
120	2 766 242	3 767 040
160	3 687 842	5 022 240
200	4 609 442	6 277 440

DR application consists of $T = 1440$ time slots, i.e., $\Delta t = 60$ s. Given the weather data from [26], we choose the 27th of March 2018, since this historical weekday comes with an average daily temperature below 5 °C, which makes it a typical day within the annual German heating period according to standard VDI 4655 [29]. Moreover, above-average solar radiation conditions prevail on the historical day under consideration, i.e., we see a power generation peak from the prosumers' PV units during noontime.

B. Implementation and Computational Setup

For the implementation of the convex optimization models and the exchange ADMM algorithm according to (15)–(20), we use the open-source *pycity_scheduling* optimization framework package [30]. Based on the homogeneous setup described in Section IV-A, the initial portfolio balancing optimization model in (7)–(9) consists of $1\,442 + 23\,040\,N$ continuous variables and $1\,440 + 31\,380\,N$ affine constraints, where N is the total number of prosumers considered inside the portfolio. Accordingly, this means that the subproblem of one single prosumer consists of 23 040 continuous variables and 31 380 affine constraints. Table I summarizes the overall size of the optimization problem in (7)–(9) for $N = \{1, 40, 80, 120, 160, 200\}$ prosumers.

Our distributed exchange ADMM algorithm implementation supports parallel computations using the message passing interface (MPI) standard and the MPI library *openmpi* [31]. We perform all MPI computations on a homogeneous *CentOS Linux* compute cluster that consists of five loosely-coupled computing nodes with 42 exclusive *Intel Xeon Platinum 8160 2.1 GHz* CPU cores and 192 GB main memory each. All MPI communication between the nodes is realized via a high bandwidth *Infiniband* network. For the runtime measurements conducted in the following subsections, we only report the pure algorithm wall-clock execution time that does not include additional computations such as the model generation from file system files, debug outputs, and other operations. In this context, we perform all runtime measurements multiple times and always report the average wall-clock execution time. For the computation of constrained optimization problems, we alternately use the commercial *Gurobi* mathematical programming solver [32] as well as the open-source *COIN-OR Ipopt* solver [33]. This is because we also want to quantify and benchmark the impact of the underlying mathematical optimization solver applied. Open-source solvers frequently show a worse computational performance compared to commercial solvers, but one can use them freely and without any licensing costs. For both *Gurobi* and *Ipopt*, we disable the console outputs and logs, and keep all other configurable solver parameters at their default values.

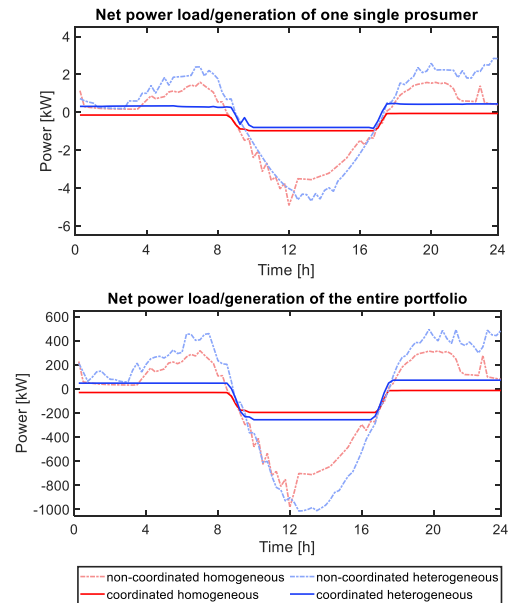


Fig. 3. Comparison of the residual net power load/generation for the non-coordinated versus coordinated case ($N = 200$).

C. Evaluation of the DR Peak Shaving Application

We first investigate the DR peak shaving application itself and show that the centralized portfolio balancing optimization problem formulation (7)–(9) satisfies the desired peak shaving intent. The goal is to quantify how the DR peak shaving application affects the load curve of single prosumers and of the entire portfolio. For this purpose, we assume a portfolio size of $N = 200$ prosumers and compare for that particular case the residual net power load/generation of a noncoordinated portfolio against a DR coordinated one. Only in the DR coordinated case, we consider both the prosumer local-level generation self-consumption and the aggregator system-level peak shaving objectives as defined in (3) and (6), respectively.

Fig. 3 shows this comparison for both one single, exemplary prosumer, and the entire portfolio. The comparison is based on i) the homogeneous portfolio setup described in Section IV-A and ii) a portfolio with a heterogeneous prosumer setup (i.e., with varying prosumer equipment) to assess the validity of our homogeneity assumption. From Fig. 3, it becomes evident that solely for the coordinated case (solid dark red and dark blue curves) the residual net power curve is flattened, because of the considered optimization objectives in line with the performed coordination of the flexible devices on the prosumers' local premises. Instead, for the noncoordinated case (dotted light red and light blue curves), consumption peaks during the morning and evening hours as well as a severe PV generation peak during noontime occur. This is what one would expect for a noncoordinated portfolio of prosumers, because of the absence of optimization goals. The schedules in light colors, hence, represent one feasible solution that satisfies all the operational constraints of the prosumers' local devices, but it does not pursue toward any optimization objective. It can also be seen from Fig. 3 that the homogeneous and heterogeneous cases differ only slightly and that they ultimately result in similar peak-shaved net power schedules for the DR coordinated case.

TABLE II
DISTRIBUTED ADMM – REQUIRED NUMBER OF ITERATIONS

Step size parameter ρ	Portfolio size					
	$N=1$	$N=40$	$N=80$	$N=120$	$N=160$	$N=200$
1.0	21	499	500	500	500	500
5.0	15	132	279	425	500	500
10.0	18	86	121	201	290	377
15.0	22	118	113	159	158	147
20.0	17	111	166	174	213	198
25.0	37	151	189	200	240	225
30.0	86	212	214	230	227	261
35.0	116	231	270	299	308	349
40.0	145	208	297	331	344	389

Besides the peak-shaved net power schedule, also the additional impact of the prosumer local-level objectives becomes apparent for the DR coordinated case: the total PV generation self-consumption share increases from 43% to 78% in this particular scenario. However, because the convex device models considered in this article only allow the prosumers for individual load shifts over time (see constraints (28) and (34) in the Appendix section), the total energy consumption of a single prosumer over optimization horizon T_h does not change for the DR coordinated scenario compared to the noncoordinated one.

D. Evaluation of the Distributed ADMM Convergence Rate

We now apply the distributed exchange ADMM algorithm to the equivalent optimization problem in (10)–(13) and investigate the convergence rate characteristics of the ADMM algorithm subject to the choice of the augmented Lagrangian penalty parameter ρ and the portfolio size N .

We define the ADMM tolerances as $\varepsilon_p = 1.0$ and $\varepsilon_d = 0.1$ and record the number of required ADMM iterations until the ADMM convergence criteria in (20) are both satisfied for the DR peak shaving application under investigation. As ADMM is known to converge to modest accuracy within a few tens of iterations for adequately defined ADMM step size parameters ρ [6], we stop the algorithm at a maximum of 500 iterations. If the algorithm would exceed 500 iterations, this implies that the ADMM step size parameter ρ is not well determined. It is worth mentioning that one can also determine the step size parameter adaptively over the ADMM iterations to improve the algorithm's speed of convergence. For instance, one may adopt the varying penalty parameter approaches from [6] or [34]. However, convergence of the ADMM algorithm in theory cannot be guaranteed in all cases if the step size parameter varies by iteration [34]. In contrast, for convex problems, the distributed ADMM always converges to the optimal solution for $k \rightarrow \infty$ when using a fixed augmented Lagrangian penalty parameter ρ . For this reason, we do not further investigate into varying penalty parameter schemes.

Table II shows the ADMM convergence rate results, i.e., the number of required iterations, for different portfolio sizes N and step size parameter choices ρ , respectively. It should be noted that the number of required iterations is the same for both mathematical optimization solvers applied in this article. For evaluation purposes, we mark the minimum number of required iterations per column in Table II in bold. From the number of iterations, it becomes evident that the choice on ρ is crucial as it

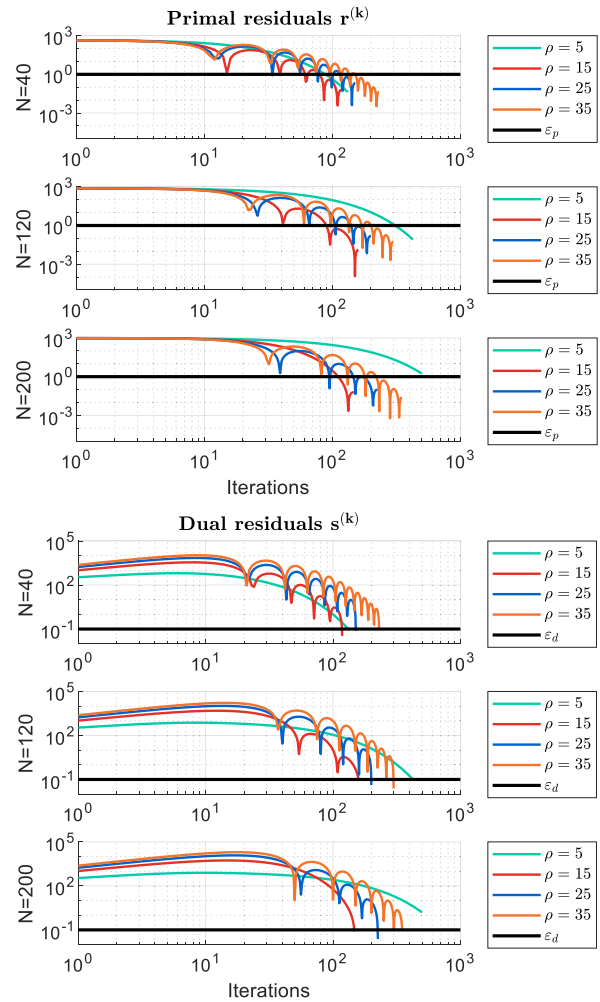


Fig. 4. Primal and dual residuals over the ADMM iterations for selected penalty parameters ρ and portfolio sizes N .

significantly affects the speed of convergence of the distributed exchange ADMM algorithm.

Fig. 4 additionally visualizes the primal and dual residuals for $\rho = \{5, 15, 25, 35\}$ as well as $N = \{40, 120, 200\}$ over the iterations using a double-logarithmic scale. Both the primal and dual residuals decrease exponentially with a growing number of ADMM iterations k . Although the descent within the first iterations is in fact very steep, a slight variation in ρ can cause a difference of hundreds of iterations for the same scenario before the primal and dual stopping criteria are finally satisfied or before the maximum of 500 iterations is reached. Boyd [6] explained this as follows: large values for ρ result in small primal residuals $r^{(k)}$ but large dual residuals $s^{(k)}$, whereas smaller values for ρ yield large primal residuals $r^{(k)}$ but small dual residuals $s^{(k)}$. Thus, as both the primal and dual residuals must take small values according to the ADMM convergence criteria in (20), a moderate, i.e., neither a too small nor too large penalty parameter ρ must be chosen.

In practice, though, the question of which specific value the augmented Lagrangian penalty parameter should ideally take is a problem-specific endeavor. For our application, we can conclude from Table II and Fig. 4 that $\rho = 15.0$ is a decent choice—in particular for large portfolio sizes N .

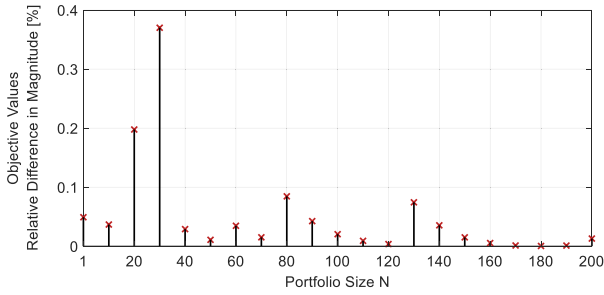


Fig. 5. Final objective value of the distributed ADMM versus the centralized reference benchmark optimization using *Gurobi*.

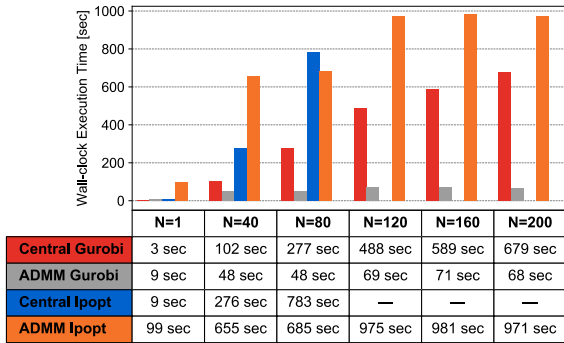


Fig. 6. Algorithmic performance of the distributed ADMM versus the centralized reference benchmark optimization.

E. Evaluation of the Distributed ADMM Optimality

Based on ADMM parameters $\rho = 15.0$, $\varepsilon_p = 1.0$, and $\varepsilon_d = 0.1$, we verify that these choices sufficiently satisfy the optimality condition of the distributed ADMM algorithm. In other words, we prove that, for the optimal exchange problem in (10)–(13), the ADMM algorithm provides almost the same optimum compared to the centralized portfolio balancing problem in (7)–(9). For this purpose, we compare the final objective value of our distributed and parallel ADMM implementation with the one of the centralized reference benchmark optimization using the *Gurobi* solver.

Fig. 5 shows the numerical relative difference in magnitude between the two finally obtained objective values for $N = \{1, 10, 20, \dots, 200\}$ prosumers. One can see that the relative objective value difference in magnitude is smaller than 0.4 % for all considered portfolio sizes. This indicates that the optimization problem solution provided by the distributed ADMM algorithm is optimal with a near-zero optimality gap compared to the centralized reference benchmark solution. For the *Ipopt* solver, we obtain a near-zero optimality gap, too.

F. Evaluation of the Distributed ADMM Performance

In this simulation experiment, we investigate the algorithmic performance of the distributed ADMM algorithm versus the centralized reference benchmark optimization subject to the portfolio size N and the optimization solver applied.

Based on ADMM parameters $\rho = 15.0$, $\varepsilon_p = 1.0$, and $\varepsilon_d = 0.1$, Fig. 6 visualizes the measured average wall-clock execution time for $N = \{1, 40, 80, 120, 160, 200\}$ prosumers. We can see from this figure that the parallel implementation for the

distributed ADMM algorithm outperforms the centralized reference benchmark optimization for a large number of prosumers N . In the case of the *Gurobi* solver, this leads to speedup factors of more than eight. The breakeven point in execution time is located around $N = 40$ prosumers for the *Gurobi* solver and around $N = 80$ for the *Ipopt* solver. This outcome is plausible, because the optimization problem size and, hence, the complexity of the centralized computation increases significantly for growing N , whereas the total execution time of the parallel ADMM algorithm predominantly depends on the number of ADMM iterations required. It is important to emphasize that a larger portfolio size does not necessarily imply an increased number of ADMM iterations. For example, Table II indicates that ADMM requires 147 iterations for $N = 200$ prosumers, but 159 iterations and 158 iterations for $N = 120$ and $N = 160$ prosumers, respectively. Accordingly, Fig. 6 shows that ADMM has a slightly shorter execution time for $N = 200$ prosumers compared to $N = 120$ and $N = 140$ prosumers.

Despite the pure execution times, another aspect becomes apparent from Fig. 6: the required main memory demand by the optimization solver becomes a challenge for large centralized optimization problems. We can identify this issue for the *Ipopt* solver for which one we are not able to perform a centralized computation for more than $N = 102$ prosumers. In this case, the *Ipopt* solver runs out of memory and, hence, it fails to compute the centralized optimization problem. In contrast, we can still solve the distributed optimization problem with the parallel ADMM implementation, because the main memory demand per MPI process of approximately 120 MB remains comparatively small. Thus, the overall main memory requirements are substantially lower in the distributed case.

G. Discussion of the Results and Limitations

The feasibility and optimality properties of the distributed ADMM algorithm for distributed DR applications that are subject to convex modeling approaches have proven satisfactory in prior research works such as in [9], [11], and [23]. Our study confirms this outcome, but clearly indicates that the right choice on the ADMM step size parameter is of utmost importance to let the algorithm converge quickly. Even for an entirely parallel implementation of ADMM, the total number of ADMM iterations required still has the most significant impact on the overall execution time of the algorithm and, hence, on the performance and scalability of the final DR application.

It is important to stress that our findings, however, are limited to portfolio sizes of up to $N = 200$ prosumers. This is because we assign one exclusive CPU computing resource to every ADMM subproblem, i.e., to every prosumer local-level optimization problem. One could also integrate $N > 200$ prosumers, but this would mean that we must run more computing processes than physical cores available, which would make the performed execution time measurements meaningless due to saturation effects (“*Amdahl’s law*”).

Moreover, we are well aware that our assumptions on a performant compute cluster and negligibly short communication delays due to ideal MPI communication are not valid in a real field application. In particular, nonideal communication over noisy channels that causes data delays and/or packet losses is common in practice and may lead to instability and poor convergence of the ADMM algorithm. Related work such as in [35]

has already successfully approached this issue by introducing communication-robust modifications to the iterative process of ADMM. In our work, however, we are particularly interested in the general trends and limitations of the best-case computational performance scenario considering it as an ideal and stable benchmark case for real-world DR applications. A comprehensive analysis of the performance of the ADMM algorithm that considers communication-robust strategies is nevertheless important and linked to future work.

Furthermore, one may argue that the total execution time for a day-ahead scheduling operation using a centralized optimization and a powerful optimization solver such as *Gurobi* is still eligible. In particular, this would be the case for optimization horizons T_h with a lower resolution than in our work, e.g., such as for $\Delta t = 15$ min up to $\Delta t = 1$ h, which is common in the electricity, heat and gas domain. The explicit choice on the time slot of the optimization problem is thereby strongly related to the tradeoff between the required level of detail and the computational effort. However, based on the performance results shown, one must oppose that, in the worst-case and independent of the optimization horizon T_h , a centralized optimization almost scales nonpolynomially and, thus, may fail for portfolios that are more complex than in our benchmark setup. For instance, a real heterogeneous portfolio setup may become more challenging to solve by both a centralized and distributed optimization compared to the homogeneous case. This is because the optimization problems of the local prosumers become slightly more complex as a result of the more diversified assets. Yet, this would only slightly affect the overall convergence rate and solution time of the ADMM algorithm: the local optimization problems remain convex and, hence, the fast convergence property of ADMM is preserved. On the other hand, in the centralized case, the global optimization problem becomes more complex in an order of magnitude of the number of prosumers considered. Therefore, as the number of prosumers N increases, it becomes increasingly challenging to succeed in solving the centralized problem, even if the overall formulation of the portfolio balancing optimization problem remains convex. Optimization horizons with a lower resolution would in fact reduce the complexity, but this approach ultimately does not fully resolve the aforementioned performance drawbacks of a purely centralized optimization for very large prosumer portfolios.

V. CONCLUSION

For the coordination and optimal control of DERs in slow-dynamics DR applications, distributed optimization approaches feature several advantages over purely centralized computations. In particular, the simulation experiments conducted in this article demonstrate that the distributed ADMM algorithm is superior compared to a centralized computation also from the computational performance point of view, which has not been sufficiently well quantified before. Thus, the consideration of the distributed ADMM algorithm can be beneficial for commercial DR aggregators that maintain large prosumer portfolios. We encourage such DR aggregators to make use of the distributed and hierarchical DR architecture proposed in this article. The architecture makes it possible to readily integrate hundreds of distributed prosumers and flexible devices in a scalable, modular, and privacy preserving way.

A centralized formulation of the portfolio balancing optimization problem would usually become too complex to solve

by DR aggregators for large prosumer portfolios. However, by reformulating the portfolio balancing optimization problem as an equivalent and separable optimal exchange problem, the distributed ADMM algorithm can be employed. ADMM overcomes the abovementioned limitation and prosumers can even be equipped with low-cost computation hardware and a comparatively slow mathematical optimization solver to solve their individual local-level ADMM subproblems. Although the ADMM algorithm guarantees convergence toward the global optimal solution for convex optimization problem formulations, nevertheless, our findings indicate that the right choice on the ADMM step size parameter is key for the overall performance of the algorithm.

One notable limitation of our findings is in the usage of the MPI standard for parallel ADMM computations, which does not reflect DR architecture conditions close to the reality, where one may need to transmit shared ADMM variables over delaying and possibly unreliable communication links. The analysis of the impact caused by the communication infrastructure is left to future work. We also want to further investigate the impact of variations in the optimization problem modeling. A centralized computation may become even more challenging for problems that contain binary/integer variables or nonconvex constraints. In this context, we also intend to evaluate, compare, and assess variants of the distributed ADMM algorithm other than exchange ADMM as well as related distributed optimization algorithms such as ALADIN.

APPENDIX

PROSUMER LOCAL DEVICES AND CONSTRAINTS

Inspired by earlier work [24], we define the local devices and optimization constraints with which a single prosumer i may be equipped in this article, i.e., we specify the convex sets of local prosumer constraints \mathbb{X}_i defined in (2). We assume that the residual load $\mathbf{P}_{el,i} \in \mathbb{X}_i$ of prosumer i consists of a finite number of inflexible loads $l \in \mathcal{L}$, PV units $p \in \mathcal{P}$, stationary battery storage units $b \in \mathfrak{B}$, and electro-thermal heat pump plus thermal energy storage units $h \in \mathcal{H}$. These devices all contribute to the net residual electric power load $\mathbf{P}_{el,i}$ of prosumer i . Thus, the aggregated net residual power load per prosumer i over all its local devices can be written as

$$\mathbf{P}_{el,i} = \sum_{l \in \mathcal{L}, p \in \mathcal{P}, b \in \mathfrak{B}, h \in \mathcal{H}} \mathbf{P}_{el,i}^l + \mathbf{P}_{el,i}^p + \mathbf{P}_{el,i}^b + \mathbf{P}_{el,i}^h \quad (21)$$

where variables $\mathbf{P}_{el,i}^l$, $\mathbf{P}_{el,i}^p$, $\mathbf{P}_{el,i}^b$, and $\mathbf{P}_{el,i}^h$ denote the power contribution of the l th inflexible load, p th PV unit, b th stationary battery storage unit, and h th electro-thermal heat pump plus thermal energy storage unit, respectively.

A. Inflexible Loads

An inflexible load $l \in \mathcal{L}$ is defined by a fixed electric power consumption profile $\mathbf{P}_{el,load,i}^l$, see constraint

$$\mathbf{P}_{el,i}^l = \mathbf{P}_{el,load,i}^l \quad (22)$$

B. PV Units

A PV unit $p \in \mathcal{P}$ is defined by a fixed electric power generation profile $\mathbf{P}_{el,gen,i}^p$. This means that we do not consider

curtailment of PV generation, see constraint

$$P_{el,i}^p = P_{el,gen,i}^p \quad (23)$$

C. Stationary Battery Storage Units

A stationary battery storage unit $b \in \mathfrak{B}$ is a flexible device that can temporarily buffer electrical energy and that is capable to provide almost continuous power charging and discharging rates. Its physical operation constraints are as follows:

$$P_{el,i}^b = P_{el,charge,i}^b - P_{el,discharge,i}^b \quad (24)$$

$$0 \leq P_{el,charge,i}^{b,t} \leq P_{el,charge,max,i}^b, \quad \forall t \in T_h \quad (25)$$

$$0 \leq P_{el,discharge,i}^{b,t} \leq P_{el,discharge,max,i}^b, \quad \forall t \in T_h \quad (26)$$

$$0 \leq E_{el,stor,i}^{b,t} \leq E_{el,stor,max,i}^b, \quad \forall t \in T_h \quad (27)$$

$$E_{el,stor,i}^{b,t=0} = E_{el,stor,i}^{b,t=T} = E_{el,stor,ini,i}^b \quad (28)$$

$$E_{el,stor,i}^{b,t} = E_{el,stor,i}^{b,t-1} k_{loss,i}^b + \left(P_{el,charge,i}^{b,t} - P_{el,discharge,i}^{b,t} \right) \Delta t, \quad \forall t \in T_h. \quad (29)$$

In (24)–(26), variable $P_{el,i}^b$ defines the net power contribution of the stationary battery storage unit. For all time steps t , it is subject to a maximum charging power $P_{el,charge,max,i}^b$ and maximum discharging power $P_{el,discharge,max,i}^b$, respectively. Variable $E_{el,stor,i}^{b,t}$ in (27) represents the state of charge of the stationary battery storage unit with maximum energy capacity $E_{el,stor,max,i}^b$. Constraint (28) defines both the initial and final state of charge of the battery storage unit for time steps $t = 0$ as well as $t = T$. The equality for the initial and final state of charge enforces the “energy neutrality” of a stationary battery storage unit over the optimization horizon T_h . Constraint (29) ensures the energy balance of the battery storage unit over T_h . The energy stored at time step t equals the energy stored in the previous time step reduced by losses $k_{loss,i}^b$ that model, e.g., self-discharging effects, plus the energy charged or discharged during the current time step. A note of caution is due here, because (29) does not include the battery storage unit’s charging and discharging efficiencies. This is because the modeling of charging/discharging efficiencies would typically result in a nonconvex, mixed-integer based problem formulation. Thus, we consider a stationary battery storage unit as an ideal device, i.e., we assume a charging and discharging efficiency of 100 % each.

D. Heat Pump Plus Thermal Energy Storage Units

An electro-thermal heat pump plus thermal energy storage unit $h \in \mathcal{H}$ represents a state-of-the-art air-to-water heat pump connected to a hot water tank buffer. Thermal energy storage decouples the prosumer’s thermal demand from the heat pump’s thermal generation and, hence, allows the heating system for thermal load shifts in time. The physical operation constraints for a heat pump plus thermal energy storage unit are as follows:

$$P_{th,i}^{h,t} = P_{el,i}^{h,t} \tilde{\eta}_{COP,i}^{h,t}, \quad \forall t \in T_h \quad (30)$$

$$0 \leq P_{th,i}^{h,t} \leq P_{th,nom,i}^h, \quad \forall t \in T_h \quad (31)$$

$$\tilde{\eta}_{COP,i}^{h,t} = \eta_{COP,nom,i}^h \frac{T_{sink,i}^{h,t}}{T_{sink,i}^{h,t} - T_{source,i}^{h,t}}, \quad \forall t \in T_h \quad (32)$$

$$0 \leq E_{th,stor,i}^{h,t} \leq E_{th,stor,max,i}^h, \quad \forall t \in T_h \quad (33)$$

$$E_{th,stor,i}^{h,t=0} = E_{th,stor,i}^{h,t=T} = E_{th,stor,ini,i}^h \quad (34)$$

$$E_{th,stor,i}^{h,t} = E_{th,stor,i}^{h,t-1} k_{loss,i}^h + \left(P_{th,i}^{h,t} - P_{th,demand,i}^{h,t} \right) \Delta t, \quad \forall t \in T_h. \quad (35)$$

In constraint (30), variable $P_{th,i}^{h,t}$ defines the thermal generation of the heat pump at time step t that is restricted by the heat pump’s nominal thermal power generation $P_{th,nom,i}^h$ in (31). The ratio between thermal generation and electric consumption of the heat pump is defined by the current Coefficient of Performance (COP) $\tilde{\eta}_{COP,i}^{h,t}$, which depends on the nominal COP denoted as $\eta_{COP,nom,i}^h$ as well as on the current difference between sink and source temperatures $T_{sink,i}^{h,t}$ and $T_{source,i}^{h,t}$, see (32). Variable $E_{th,stor,i}^{h,t}$ in (33) represents the state of charge of the thermal energy storage unit with maximum energy capacity $E_{th,stor,max,i}^h$. Constraint (34) defines both the initial and final state of charge of the thermal energy storage. In constraint (35), the thermal energy stored in the hot water tank buffer at time step t equals the energy stored in the previous time step reduced by losses $k_{loss,i}^h$ that model, e.g., temperature losses, plus the thermal energy charged, and/or discharged during the current time step. Thermal charging thereby corresponds to the heat pump’s thermal generation $P_{th,i}^{h,t}$, whereas discharging corresponds to the prosumer’s thermal demand $P_{th,demand,i}^{h,t}$.

REFERENCES

- [1] G. Lipari, G. Del Rosario, C. Corchero, F. Ponci, and A. Monti, “A real-time commercial aggregator for distributed energy resources flexibility management,” *Sustain. Energy, Grids Netw.*, vol. 15, pp. 63–75, 2018.
- [2] J. Naughton, H. Wang, S. Riaz, M. Cantoni, and P. Mancarella, “Optimization of multi-energy virtual power plants for providing multiple market and local network services,” *Elect. Power Syst. Res.*, vol. 189, 2020, Art. no. 106775.
- [3] D. K. Molzahn et al., “A survey of distributed optimization and control algorithms for electric power systems,” *IEEE Trans. Smart Grid*, vol. 8, no. 6, pp. 2941–2962, Nov. 2017.
- [4] V. Omel'chenko and V. Manokhin, “Optimal balancing of wind parks with virtual power plants,” *Front. Energy Res.*, vol. 9, 2021, Art. no. 665295.
- [5] The edgeFLEX 2020 consortium, “Providing Flexibility to the grid by enabling VPPs to offer both fast and slow dynamics control services,” edgeflex-h2020.eu, Accessed: Jan. 13, 2023. [Online]. Available: <https://edgeflex-h2020.eu/about.html>
- [6] S. Boyd, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *FNT Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.
- [7] A. Soares et al., “Distributed optimization algorithm for residential flexibility activation—Results from a field test,” *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 4119–4127, Sep. 2019.
- [8] S. Huang, Y. Sun, and Q. Wu, “Stochastic economic dispatch with wind using versatile probability distribution and L-BFGS-B based dual decomposition,” *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6254–6263, Nov. 2018.
- [9] J. Rivera, C. Goebel, and H.-A. Jacobsen, “Distributed convex optimization for electric vehicle aggregators,” *IEEE Trans. Smart Grid*, vol. 8, no. 4, pp. 1852–1863, Jul. 2017.

- [10] X. Zhou, S. Zou, P. Wang, and Z. Ma, "ADMM-based coordination of electric vehicles in constrained distribution networks considering fast charging and degradation," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 565–578, Jan. 2021.
- [11] S. Faddel, Q. Z. Zun, and G. Tian, "Modeling and coordination of commercial buildings in distribution systems," *IEEE Trans. Ind. Appl.*, vol. 58, no. 2, pp. 1657–1663, Mar./Apr. 2022.
- [12] S. D. Manshadi, G. Liu, M. E. Khodayar, J. Wang, and R. Dai, "A distributed convex relaxation approach to solve the power flow problem," *IEEE Syst. J.*, vol. 14, no. 1, pp. 803–812, Mar. 2020.
- [13] A. Attarha, P. Scott, and S. Thiébaux, "Affinely adjustable robust ADMM for residential DER coordination in distribution networks," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1620–1629, Mar. 2020.
- [14] Q. Li et al., "Parallel and distributed optimization method with constraint decomposition for energy management of microgrids," *IEEE Trans. Smart Grid*, vol. 12, no. 6, pp. 4627–4640, Nov. 2021.
- [15] A. Kargarian et al., "Toward distributed/decentralized DC optimal power flow implementation in future electric power systems," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 2574–2594, Jul. 2018.
- [16] S. Lilla, C. Orozco, A. Borghetti, F. Napolitano, and F. Tossani, "Day-ahead scheduling of a local energy community: An alternating direction method of multipliers approach," *IEEE Trans. Power Syst.*, vol. 35, no. 2, pp. 1132–1142, Mar. 2020.
- [17] P. Olivella-Rosell et al., "Centralised and distributed optimization for aggregated flexibility services provision," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3257–3269, Jul. 2020.
- [18] X. Kou et al., "A scalable and distributed algorithm for managing residential demand response programs using alternating direction method of multipliers (ADMM)," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 4871–4882, Nov. 2020.
- [19] A. Singh, B. K. Sethi, D. Singh, and R. K. Misra, "Shapley value method and stochastic Dantzig–Wolfe decomposition for decentralized scheduling of multimicrogrid," *IEEE Syst. J.*, vol. 16, no. 2, pp. 2672–2683, Jun. 2022.
- [20] A. Engelmann, Y. Jiang, T. Mühlpfordt, B. Houska, and T. Faulwasser, "Toward distributed OPF using ALADIN," *IEEE Trans. Power Syst.*, vol. 34, no. 1, pp. 584–594, Jan. 2019.
- [21] Expert Group 3 of the Smart Grids Task Force, "Regulatory recommendations for the deployment of flexibility: SGTF-EG3 report," 2015. Accessed: Jan. 13, 2023. [Online]. Available: https://projects.2014-2020.interregeurope.eu/fileadmin/user_upload/tx_tevprojects/library/EG3%20Final%20-%20January%202015.pdf
- [22] M. Diekerhof, F. Peterssen, and A. Monti, "Hierarchical distributed robust optimization for demand response services," *IEEE Trans. Smart Grid*, vol. 9, no. 6, pp. 6018–6029, Nov. 2018.
- [23] M. Juelsgaard, "Utilizing distributed resources in smart grids—A coordination approach," Ph.D. dissertation, Dept. Electron. Syst. Automat. Control, Aalborg Univ., Aalborg, Denmark, 2014.
- [24] M. Diekerhof, S. Schwarz, F. Martin, and A. Monti, "Distributed optimization for scheduling electrical demand in complex city districts," *IEEE Syst. J.*, vol. 12, no. 4, pp. 3226–3237, Dec. 2018.
- [25] Institut Wohnen und Umwelt GmbH and the Consortium of EU IEE Project EPISCOPE, "National building typologies," episcope.eu. Accessed: Jan. 13, 2023. [Online]. Available: <https://episcope.eu/building-typology/>
- [26] Deutscher Wetterdienst (DWD), "Open data server of the German meteorological service (DWD)," German, Apr. 2022. Accessed: Jan. 13, 2023. [Online]. Available: <https://opendata.dwd.de/>
- [27] W. Zhou, H. Yang, and Z. Fang, "A novel model for photovoltaic array performance prediction," *Appl. Energy*, vol. 84, no. 12, pp. 1187–1198, 2007.
- [28] Bundesverband der Energie- und Wasserwirtschaft e.V. (BDEW), "German standard load profiles," German, Jan. 2017. Accessed: Jan. 13, 2023. [Online]. Available: <https://www.bdew.de/media/documents/Profile.zip>
- [29] *Reference Load Profiles of Single-Family and Multi-Family Houses for the Use of CHP Systems*, VDI 4655:2008-05, German Verein Deutscher Ingenieure e.V. (VDI), German, Berlin, Germany, May 2008.
- [30] S. Schwarz, S. A. Uerlich, and A. Monti, "Pycity_Scheduling—A Python framework for the development and assessment of optimisation-based power scheduling algorithms for multi-energy systems in city districts," *SoftwareX*, vol. 16, 2021, Art. no. 100839.
- [31] Open MPI v3.1.3, 2018. [Online]. Available: <https://www.open-mpi.org/software/ompi/v3.1/>
- [32] Gurobi Optimization LCC, "Gurobi optimizer 9.1.1," 2020. [Online]. Available: <https://www.gurobi.com/documentation/9.1/>
- [33] COIN-OR Foundation, "Ipop3.12.13," 2019. [Online]. Available: <https://www.coin-or.org/download/source/Ipop/>
- [34] M. Kraning, E. Chu, J. Lavaei, and S. Boyd, "Dynamic network energy management via proximal message passing," *FNT Optim.*, vol. 1, no. 2, pp. 73–126, 2014.
- [35] X. Liang, Z. Li, W. Huang, Q. H. Wu, and H. Zhang, "Relaxed alternating direction method of multipliers for hedging communication packet loss in integrated electrical and heating system," *J. Modern Power Syst. Clean Energy*, vol. 8, no. 5, pp. 874–883, 2020.



Sebastian Schwarz (Member, IEEE) received the B.Sc. and M.Sc. (with Hons.) degree in electrical engineering from RWTH Aachen University, Aachen, Germany, in 2015 and 2018, respectively.

He is currently a Research Associate with the E.ON Energy Research Center, Institute for Automation of Complex Power Systems, RWTH Aachen University. His research interests include intelligent energy management solutions and distributed optimization approaches for multienergy local balancing systems.

Mr. Schwarz is an Executive Committee Member of the IEEE PES Germany Chapter. He was the recipient of the 2019 IEEE PES Germany Chapter Best Master Thesis Award and the 2020 IEEE PES Outstanding Chapter Volunteer Award.



Antonello Monti (Senior Member, IEEE) received the M.Sc. (summa cum laude) and Ph.D. degrees in electrical engineering from the Politecnico di Milano, Milan, Italy, in 1989 and 1994, respectively.

He started his career with Ansaldo Industria and then joined Politecnico di Milano as an Assistant Professor, in 1995. In 2000, he joined the Department of Electrical Engineering, University of South Carolina, Columbia, SC, USA, as an Associate Professor and later became a Full Professor. Since 2008, he has been the Director of the Institute for Automation

of Complex Power Systems, E.ON Energy Research Center, RWTH Aachen University, Aachen, Germany. Since 2019, he holds a double appointment with Fraunhofer FIT, where he is developing the new Center for Digital Energy Aachen, Aachen, Germany. He has authored or coauthored of more than 400 peer-reviewed papers published in international journals and in the proceedings of international conferences.

Prof. Monti is an Associate Editor for *IEEE Electrification Magazine*, and a Member of the Editorial Board of the *SEGAN Journal* (Elsevier) and the Founding Board of the *Energy Informatics journal* (Springer). He was a recipient of the 2017 IEEE Innovation in Societal Infrastructure Award.