

Copyright Notice

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

Manuscript submitted to IEEE PowerTech 2025 conference (<https://2025.ieee-powertech.org/>).

Scheduling of aggregated energy systems using a distributed augmented Lagrangian based alternating direction inexact Newton optimization algorithm

1st Filip Sobic

Department of Energy
Politecnico di Milano
Milan, Italy
filip.sobic@polimi.it

2nd Sebastian Schwarz

E.ON ERC ACS
RWTH Aachen University
Aachen, Germany
sschwarz@eonerc.rwth-aachen.de

3rd Emanuele Martelli

Department of Energy
Politecnico di Milano
Milan, Italy
emanuele.martelli@polimi.it

4th Antonello Monti

E.ON ERC ACS
RWTH Aachen University
Aachen, Germany
amonti@eonerc.rwth-aachen.de

Abstract—This work investigates the suitability and applicability of the derivative-free version of the augmented Lagrangian based alternating direction inexact Newton (ALADIN) optimization algorithm for distributed weekly scheduling in aggregated energy systems. A hierarchical setup is considered, in which an aggregator acts as a system-level coordinator with a peak-shaving objective, while prosumers seek to minimize their time-of-use electricity costs. The prosumers are modeled with both flexible and inflexible electrical and thermal components, including photovoltaic systems, batteries, heat pumps, and thermal energy storage. In this scenario, the proposed ALADIN algorithm is applied to a focused test case featuring prosumer portfolios of varying sizes. Moreover, the impact of three key algorithmic parameters on ALADIN's performance is systematically analyzed. The results show that the algorithm may converge quickly, producing weekly schedules and objective values identical to those of a centralized reference benchmark optimization. However, the coupling quadratic problem of the ALADIN algorithm emerges as a critical step, with computing times being up to 33.6 times longer than those required for solving individual prosumer subproblems. This limitation reveals challenges in scalability for large-scale system setups.

Index Terms—Aggregated energy system, Augmented Lagrangian based alternating direction inexact Newton (ALADIN) algorithm, Distributed optimization, Prosumer, Scheduling

I. INTRODUCTION

Aggregated energy systems, such as those in energy communities, have gained significant research interest over the past decade due to their ability to aggregate, coordinate, and optimize a large number of heterogeneous prosumers with multi-energy demands (e.g., electricity, gas, heating, and cooling) [1]. These systems typically integrate both flexible devices (e.g., heat pumps) and non-flexible devices (e.g., photovoltaic (PV) units), along with storage technologies (e.g., battery and thermal energy storage).

To ensure efficient and grid-friendly operation of aggregated energy systems, optimization strategies are required. These strategies should determine the best possible operation schedule for flexible devices, optimizing toward a predefined objective function while satisfying the energy demands of each local prosumer and adhering to all technical constraints of the system. The most straightforward approach is centralized

optimization, which assumes that a central entity, such as an aggregator, has complete knowledge of the system, including energy demands, installed devices, and operational parameters of every prosumer [2], [3]. Although centralized formulations can be solved to global optimality even for the most general class of mixed integer nonlinear problems (MINLP) [4], they have two critical drawbacks: i) they require full knowledge of prosumer data, raising data privacy concerns, and ii) as the number of prosumers/devices increases, the problem may become computationally intractable, limiting scalability.

To address these limitations, distributed optimization approaches have been proposed. In this context, this work aims to explore and analyze the potential of the recently developed ALADIN optimization algorithm for optimal scheduling problems in aggregated energy system setups.

A. Distributed Optimization – State of the Art

State-of-the-art distributed optimization approaches include dual decomposition (DD) [5], alternating direction method of multipliers (ADMM) [6], and Dantzig-Wolfe decomposition [7]. These algorithms are well-suited for scheduling problems in aggregated energy systems, as they can exploit the separable structure of the underlying optimization problem. Specifically, they decompose the holistic problem into smaller subproblems: one for each prosumer with its flexible devices at the local level, and one for the aggregator, which pools the flexibility and energy demand/supply at the system level. These subproblems are then solved iteratively subject to penalty terms until predefined convergence criteria are satisfied, yielding the (near-)global optimum.

Work [5], for instance, studies distributed optimization for an aggregated energy system using the DD algorithm. The DD algorithm uses shadow price signals to drive the consumption/production of a common good towards equilibrium [6]. The study focuses on aggregating residential consumers equipped with storage and two adjustable loads — one characterized by specified consumption patterns, and one by a dissatisfaction function. The authors show that the DD algorithm can converge to global optimality even when using outdated

Lagrange multipliers due to lost communication messages between iterations. However, DD requires strict convexity of the objective function, limiting its applicability.

To overcome this limitation, the ADMM algorithm was proposed. ADMM combines the decomposability of DD with the superior convergence properties of the method of multipliers [6]. Work [4] makes use of ADMM for the day-ahead scheduling of aggregated energy systems comprising residential prosumers. The authors benchmark it against a centralized formulation in terms of computational times and scalability for a varying number of prosumers. The results indicate that ADMM outperforms the centralized formulation when the number of prosumers exceeds 40.

Dantzig-Wolfe decomposition is an algorithm that can solve optimization problems in a distributed way using column generation. However, it is limited to the class of (integer) linear problems. The authors in [7] use the Dantzig-Wolfe decomposition to minimize the electricity purchase cost of an aggregated energy system consisting of residential consumers characterized by uncontrollable loads, curtailable loads, uninterruptible loads, deferrable loads, thermal loads, and batteries. Their results show that the proposed formulation is able to converge to global optimality and scales better with increasing problem sizes compared to a centralized formulation.

More recently, the ALADIN algorithm has been introduced, capable of solving both convex and non-convex problems [8]. This distributed optimization algorithm exploits second-order derivative information to achieve quadratic or even superlinear convergence rates. Leveraging an inexact Newton method, the derivative-free version of the algorithm thereby reduces the computational burden by calculating Hessian matrices and gradients only approximately.

B. Research Motivation for Investigating ALADIN

Well-explored distributed optimization algorithms are tailored to specific problem classes: DD for strictly convex problems, ADMM for convex problems, and Dantzig-Wolfe decomposition for linear problems. In contrast, ALADIN is designed to tackle the most general class of MINLP problems. This capability is particularly valuable for optimizing aggregated energy systems, as it enables the inclusion of, for example, higher-order objective functions or integer variables in optimization problem formulations.

The authors in [9] use ALADIN to solve non-convex AC optimal power flow problems in a distributed fashion. The work demonstrates that ALADIN's locally quadratic convergence significantly reduces the number of iterations compared to ADMM for test cases involving 5-300 buses. Moreover, work [10] extends ALADIN to handle MINLP problems and applies it to battery scheduling. The results reveal that the MINLP reformulation outperforms the real-valued non-convex formulation of ALADIN in terms of convergence time and iteration count, although a general convergence proof has not been provided. Furthermore, work [11] introduces ALADIN- α , an open-source MATLAB implementation of ALADIN, incorpo-

rating meaningful extensions to reduce communication and coordination overhead of the algorithm.

Despite these valuable contributions, a comprehensive literature review reveals that contributions on ALADIN are rare and that no work has yet investigated the suitability of the ALADIN algorithm for complex, large-scale scheduling problems in aggregated energy systems. In this paper, we address this gap by deploying and testing ALADIN, placing particular emphasis on its convergence and scalability behavior.

C. Contributions

We adapt a derivative-free formulation of ALADIN, as described in [12], for weekly scheduling of the aggregated energy system across three scenarios involving 10, 50, and 100 prosumers. A prototype implementation of this derivative-free version of the ALADIN algorithm is made available as part of our open-source `pycity_scheduling` (v1.4.0)¹ Python project [13]. We evaluate the algorithm's performance with respect to the number of prosumers and examine the impact of three key algorithmic parameters on the number of iterations required to reach convergence.

II. PROBLEM FORMULATION

The general formulation of the scheduling problem for an aggregated energy system comprising of one aggregator and N prosumers is provided in (1). This problem is also known as portfolio balancing optimization [4].

$$\min_{x,q} f(x,q) = \sum_{i=0}^N f_i(x_i, q_i) \quad (1a)$$

$$s.t. \sum_{i=0}^N A_i x_i = 0 \quad (1b)$$

$$g_i(x_i, q_i) = 0 \quad \forall i = 0, 1, \dots, N \quad (1c)$$

$$h_i(x_i, q_i) \leq 0 \quad \forall i = 0, 1, \dots, N \quad (1d)$$

Here, it is assumed that the objective function of the overall system can be decomposed into $N + 1$ independent objective functions, where $f_0(x_0, q_0)$ refers to the objective function of the aggregator, while $f_i(x_i, q_i)$ with $i = 1, \dots, N$ are the objective functions of the prosumers. In this context, x_i are the variables of the subproblem i that are present in constraint (1b), coupling the subproblem of the aggregator with the subproblems of all prosumers. Equations (1c) and (1d) represent the local equality and inequality constraints of the subproblem i , respectively, which both depend on the coupling variables (x_i) and local variables (q_i) of the subproblem. Since this work investigates the scheduling problem of an aggregated energy system of prosumers, variables x_i for $i = 1, \dots, N$ represent the electrical power contributions by the prosumers at each time step t within the optimization horizon \mathcal{T} . Vice versa, variable x_0 is the power contribution by the aggregator to balance the aggregated energy system at the system level.

¹link to repository: https://git.rwth-aachen.de/acs/public/simulation/pycity_scheduling/-/releases/v1.4.0

Although problem (1) is a MINLP problem in the general case, we simplify the formulation in this work by assuming $x_i, q_i \in \mathbb{R}$, and $f_i(x_i, q_i)$ to be convex. This simplification enables direct and fair comparison of ALADIN with other well-suited distributed optimization algorithms, such as ADMM.

III. DERIVATIVE-FREE ALADIN ALGORITHM

This section provides the mathematical derivation of the derivative-free version of the ALADIN algorithm, as described in [12], tailored to the optimization of aggregated energy systems formulated in (1). The algorithm is modified adding a penalty parameter $\rho \geq 0$ and the termination condition (3b) from [8]. The iterative steps of the algorithm are as follows:

- 1) Decompose the original problem (1) into smaller subproblems for the aggregator and every prosumer. The idea in (2a)-(2d) is to minimize the objective per subproblem i in parallel, taking into account the feedback from the coupling quadratic program (QP) in (4). Each subproblem includes parameters λ^* and x_i^* , representing the numerical values of the dual variable of the decomposed coupling constraint and coupling variable, respectively, as computed in the previous iteration of the algorithm in step 5.

$$\min_{x_i, q_i} f_i(x_i, q_i) + \lambda^{*T} A_i x_i + \frac{\rho}{2} \|x_i - x_i^*\|_{\Sigma_i}^2 \quad (2a)$$

$$s.t. \quad g_i(x_i, q_i) = 0 \quad (2b)$$

$$h_i(x_i, q_i) \leq 0 \quad (2c)$$

$$x_i \in \mathbb{R}, \quad q_i \in \mathbb{R} \quad (2d)$$

Notice that Σ_i is a (positive semi-definite) scaling matrix for the Euclidean norm², penalizing the deviation of the new solution of x_i versus x_i^* . As suggested in [12], this matrix is set equal to H_i (cf. step 3).

- 2) Check if the termination criteria in (3a)-(3b) are satisfied to determine whether the algorithm has converged. Criterion (3a) evaluates whether the difference between solutions from consecutive iterations is within a specified tolerance, while criterion (3b) verifies whether the decomposed coupling constraint is satisfied.

$$\|x - x^*\|_2 \leq \epsilon \quad (3a)$$

$$\left\| \sum_{i=0}^N A_i x_i \right\|_2 \leq \epsilon \quad (3b)$$

Here, it applies $x = [x_0, x_1, \dots, x_N]$ as well as $x^* = [x_0^*, x_1^*, \dots, x_N^*]$, and ϵ is a user-defined numerical tolerance.

- 3) Choose positive-definite matrices H_i and compute vectors $g_i = H_i(x_i^* - x_i) - A_i^T \lambda^*$. Since the optimization problem for aggregated energy systems typically does not include cross-time-step variable multiplications in

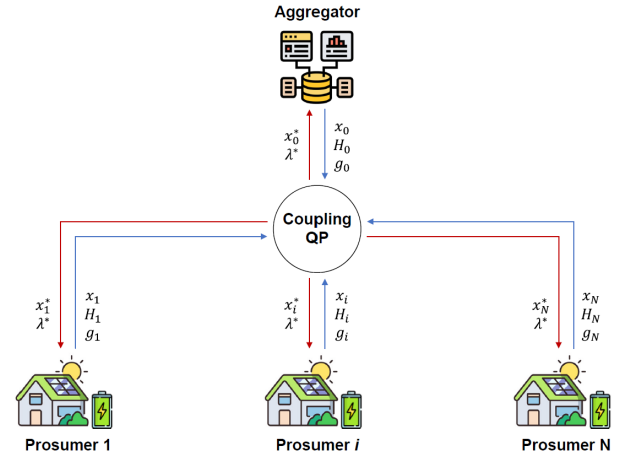


Fig. 1. Schematic of the ALADIN algorithm workflow.

the objective function, H_i can be chosen as a multiple of the identity matrix for computational effectiveness.

- 4) Solve the coupling QP in (4a)-(4b) by adjusting the solutions x_i of the individual subproblems from step 1, using variables Δx_i such that the original coupling constraint is satisfied.

$$\min_{\Delta x} \sum_{i=0}^N \left\{ \frac{1}{2} \Delta x_i^T H_i \Delta x_i + g_i^T \Delta x_i \right\} \quad (4a)$$

$$s.t. \quad \sum_{i=0}^N A_i (x_i + \Delta x_i) = 0 \quad | \quad \lambda \quad (4b)$$

- 5) Update the values of parameters x_i^* and λ_i^* in (5a)-(5b) for use in the next iteration, where α is a parameter determining the update step-size.

$$x_i^* = x_i + \alpha \Delta x_i \quad \forall i = 0, 1, \dots, N \quad (5a)$$

$$\lambda^* = \lambda^* + \alpha(\lambda - \lambda^*) \quad (5b)$$

Fig. 1 shows a schematic representation of the ALADIN algorithm workflow, highlighting what information is exchanged and in which directions.

A. Implementation Specifics

The matrices H_i grow quadratically with the number of time steps in the optimization horizon, and their count is equal to the number of subproblems considered. For large-scale optimization problems, such as weekly scheduling with many prosumers, updating H_i and solving the coupling QP thus becomes computationally expensive. To address this issue, two strategies are proposed for the efficient implementation of the derivative-free ALADIN algorithm as follows:

- 1) Since H_i are chosen as a multiple of the identity matrix, their diagonals can be stored as vectors, significantly reducing the number of parameters updated per iteration.
- 2) By using the same numerical value for H_i across all iterations, the parameters can be initialized and fixed at the start of the algorithm, further reducing the computational effort for constructing the coupling QP.

²Notation: it applies $\|x_i - x_i^*\|_{\Sigma_i}^2 = (x_i - x_i^*)^T \Sigma_i (x_i - x_i^*)$.

IV. CASE STUDY

This section demonstrates the application of the derivative-free ALADIN algorithm to distributed optimization of aggregated energy systems. Three simulation scenarios are evaluated to assess the performance of the algorithm under varying conditions. The influence of key algorithmic parameters, including matrices H_i , step-size α , and penalty parameter ρ , is systematically analyzed to determine their impact on ALADIN convergence and scalability behavior.

A. Overview

The case study focuses on a hierarchical aggregated energy system comprising an aggregator and multiple prosumers. The aggregator acts as a system-level coordinator with a peak-shaving objective, while the prosumers aim to minimize their individual time-of-use electricity costs.

Three weekly scheduling scenarios are investigated, each with a different number of prosumers: S1) 10 prosumers, S2) 50 prosumers, and S3) 100 prosumers.

In all scenarios, the prosumers are modeled as residential single-family homes (SFH) with inflexible electrical and thermal energy demands. These SFH demands are heterogeneous to represent the national building typologies of Germany, reflecting varying building construction years and energy efficiency levels. The data is derived from [14].

In addition, each SFH is equipped with a PV unit, stationary battery storage, a heat pump unit, and thermal energy storage. The modeling of these devices is described in detail in Section IV-C and they are parameterized based on publicly available data sources, including [14], [15], [16].

The optimization horizon of the weekly scheduling scenarios is divided into 15-minute intervals ($\Delta t = 15$ min), resulting in 672 discrete time steps $t \in \mathcal{T}$. The simulation is conducted for a typical German week in early spring, reflecting both moderate seasonal energy demands and PV generation.

B. Mathematical Modeling - Objective Functions

For the sake of exemplification, the aggregator's objective is defined as peak-shaving in this work, which aims to reduce the net power peaks of the aggregated energy system at the common point of coupling with the external power grid. Peak-shaving is modeled as a quadratic objective function, which can be written as follows:

$$f_0(x_0) = f_0(P_{el,0}) = \sum_{t \in \mathcal{T}} (P_{el,0}^t)^2 \quad (6)$$

where variable $P_{el,0}^t$ represents the electrical power contribution by the aggregator to supply the aggregated energy system at time step t within the optimization horizon \mathcal{T} .

For the prosumers, this work further assumes that all prosumers adopt time-of-use electricity cost minimization as their individual objective function. This objective is modeled as the following linear function:

$$f_i(x_i) = f_i(P_{el,i}) = \sum_{t \in \mathcal{T}} \bar{c}_{el,i}^t P_{el,i}^t \quad (7)$$

where variable $P_{el,i}^t$ denotes the electrical power imported or exported by the i -th prosumer at time step t within the optimization horizon \mathcal{T} , and $\bar{c}_{el,i}^t$ represents the electricity price at the same time step. Time-of-use electricity cost minimization captures the financial incentives for prosumers to optimize their electricity consumption patterns based on dynamic electricity price tariffs, encouraging cost-effective scheduling decisions for their flexible devices. The tariffs may originate from, for example, the aggregator based on contractual agreements with the prosumers.

C. Mathematical Modeling - Prosumer Constraints

Each prosumer i is assumed to have inflexible electrical demand $\bar{P}_{el,i}^{dem,t}$ and thermal demand $\bar{P}_{th,i}^{dem,t}$, and is equipped with a PV unit $p \in \mathcal{P}_i$, a battery storage system $b \in \mathcal{B}_i$, a heat pump unit $h \in \mathcal{H}_i$, and a thermal energy storage system $s \in \mathcal{S}_i$. The mathematical modeling of each of these devices is presented below. Please note that all parameters (inputs to the model) in the constraints are indicated by an overline. The numerical values for those parameters are provided in Table I. Moreover, all optimization variables are defined over \mathbb{R} , whereas all parameters are defined greater or equal to zero, unless stated otherwise.

1) *Power balances*: Power balances, represented by (8), must be satisfied for each energy type within the system. The electrical power balance, defined by (8a), ensures that the power $P_{el,i}^t$ that is imported/exported by prosumer i at time step t is equal to the sum of its inflexible electrical demand and the power contributions from all local devices at that time step. Similarly, the thermal power balance in (8b) ensures that the sum of thermal power contributions from heat pumps and thermal energy storages as well as the prosumer's thermal demand equals zero.

$$P_{el,i}^t = \bar{P}_{el,i}^{dem,t} - \sum_{p \in \mathcal{P}_i} P_{el,p}^t + \sum_{b \in \mathcal{B}_i} P_{el,b}^t + \sum_{h \in \mathcal{H}_i} P_{el,h}^t \quad \forall t \in \mathcal{T} \quad (8a)$$

$$0 = \bar{P}_{th,i}^{dem,t} - \sum_{h \in \mathcal{H}_i} P_{th,h}^t + \sum_{s \in \mathcal{S}_i} P_{th,s}^t \quad \forall t \in \mathcal{T} \quad (8b)$$

2) *PV units*: A PV unit $p \in \mathcal{P}_i$ is assumed to be non-curtailable and thus its assumed power generation equals a PV production forecast $\bar{P}_{el,p}^{t,forc}$ as follows:

$$P_{el,p}^t = \bar{P}_{el,p}^{t,forc} \quad \forall t \in \mathcal{T} \quad (9)$$

3) *Battery storage systems*: The behavior of each battery storage system $b \in \mathcal{B}_i$ is modeled by the constraints in (10). Constraint (10a) divides the battery power contribution in the electrical power balance (8a) into charging ($P_{el,b}^{ch,t}$) and discharging ($P_{el,b}^{disch,t}$) components, upper bounded between by maximum technical limits (parameters $\bar{P}_{el,b}^{ch,max}$ and $\bar{P}_{el,b}^{disch,max}$ in constraints (10b) and (10c), respectively). Moreover, the battery's energy level $E_{el,b}^t$ is upper bounded by its maximum capacity $\bar{E}_{el,b}^{max}$, as described in constraint (10d). The battery's energy level is updated according to the balance equation in (10e), which incorporates the charging and

discharging efficiencies $\bar{\eta}_b^{ch}$ and $\bar{\eta}_b^{disch}$, as well as the time step duration Δt . Additionally, initial and final energy levels of the battery are specified in constraints (10f) and (10g).

$$P_{el,b}^t = P_{el,b}^{ch,t} - P_{el,b}^{disch,t} \quad \forall t \in \mathcal{T} \quad (10a)$$

$$0 \leq P_{el,b}^{ch,t} \leq \bar{P}_{el,b}^{ch,max} \quad \forall t \in \mathcal{T} \quad (10b)$$

$$0 \leq P_{el,b}^{disch,t} \leq \bar{P}_{el,b}^{disch,max} \quad \forall t \in \mathcal{T} \quad (10c)$$

$$0 \leq E_{el,b}^t \leq \bar{E}_{el,b}^{max} \quad \forall t \in \mathcal{T} \quad (10d)$$

$$E_{el,b}^t = E_{el,b}^{t-1} + \left(\bar{\eta}_b^{ch} P_{el,b}^{ch,t} - \frac{P_{el,b}^{disch,t}}{\bar{\eta}_b^{disch}} \right) \Delta t \quad \forall t \in \mathcal{T}, t \geq 1 \quad (10e)$$

$$E_{el,b}^{t=0} = \bar{E}_{el,b}^{ini} \quad (10f)$$

$$E_{el,b}^{t=671} = \bar{E}_{el,b}^{final} \quad (10g)$$

4) *Heat pumps*: The behavior of each heat pump $h \in \mathcal{H}_i$ is modeled by the constraints in (11). A heat pump's electrical power consumption $P_{el,h}^t$ is related to its thermal power production $P_{th,h}^t$ via the coefficient of performance ($\bar{\eta}_{COP,h}^t$), which depends on the heat pump's quality grade $\bar{\eta}_h^{quality}$ and the temperature difference between the outdoor ($\bar{T}_{outdoor}^t$) and the sink temperature (\bar{T}_{sink}), see (11a) and (11c). The heat pump's thermal power production is limited by its nominal thermal power capacity $\bar{P}_{th,h}^{nom}$, as described in constraint (11b).

$$P_{el,h}^t = \frac{1}{\bar{\eta}_{COP,h}^t} P_{th,h}^t \quad \forall t \in \mathcal{T} \quad (11a)$$

$$0 \leq P_{th,h}^t \leq \bar{P}_{th,h}^{nom} \quad \forall t \in \mathcal{T} \quad (11b)$$

$$\bar{\eta}_{COP,h}^t = \bar{\eta}_h^{quality} \cdot \frac{\bar{T}_{sink}}{\bar{T}_{sink} - \bar{T}_{outdoor}^t} \quad \forall t \in \mathcal{T} \quad (11c)$$

5) *Thermal energy storage systems*: A thermal energy storage system $s \in \mathcal{S}_i$ is modeled by similar constraints to those of a battery storage system, with some notable differences. First, there is no need to separate charging and discharging power contributions in the thermal power balance ($P_{th,s}^t$), as thermal storage is assumed to have no charging/discharging efficiencies. Instead, the evolution of the thermal storage energy level in (12a) integrates a thermal loss coefficient $\bar{\eta}_s^{loss}$, accounting for thermal energy losses over time. Second, it is assumed that there is no practical limit for a thermal energy storage system on the maximum power that can be charged or discharged per time step, thus excluding those constraints.

$$E_{th,s}^t = \bar{\eta}_s^{loss} E_{th,s}^{t-1} + P_{th,s}^t \Delta t \quad \forall t \in \mathcal{T}, t \geq 1 \quad (12a)$$

$$0 \leq E_{th,s}^t \leq \bar{E}_{th,s}^{max} \quad \forall t \in \mathcal{T} \quad (12b)$$

$$E_{th,s}^{t=0} = \bar{E}_{th,s}^{ini} \quad (12c)$$

$$E_{th,s}^{t=671} = \bar{E}_{th,s}^{final} \quad (12d)$$

TABLE I
MODEL PARAMETER VALUES

model	parameter	value	model	parameter	value
$b \in \mathcal{B}_i$	$\bar{P}_{el,b}^{ch,max}$	4.6 kW	$h \in \mathcal{H}_i$	$\bar{\eta}_h^{quality}$	36 %
$b \in \mathcal{B}_i$	$\bar{P}_{el,b}^{disch,max}$	4.6 kW	$h \in \mathcal{H}_i$	\bar{T}_{sink}	328.15 K
$b \in \mathcal{B}_i$	$\bar{E}_{el,b}^{max}$	13.5 kWh	$h \in \mathcal{H}_i$	$\bar{P}_{th,h}^{nom}$	8 - 16 kW
$b \in \mathcal{B}_i$	$\bar{\eta}_b^{ch}$	95 %	$s \in \mathcal{S}_i$	$\bar{E}_{th,s}^{max}$	2 - 4 kWh
$b \in \mathcal{B}_i$	$\bar{\eta}_b^{disch}$	95 %	$s \in \mathcal{S}_i$	$\bar{\eta}_s^{loss}$	99 %
$b \in \mathcal{B}_i$	$\bar{E}_{el,b}^{ini}$	50 %	$s \in \mathcal{S}_i$	$\bar{E}_{th,s}^{ini}$	50 %
$b \in \mathcal{B}_i$	$\bar{E}_{el,b}^{final}$	50 %	$s \in \mathcal{S}_i$	$\bar{E}_{th,s}^{final}$	50 %

D. Simulation Setup and Derivative-Free ALADIN Parameters

All simulations are conducted on an AMD EPYC 7301 compute cluster featuring 128 cores and 512 GB of memory. The commercial solver Gurobi 12.0 [17] is used to solve the individual subproblems as well as the coupling QP problem. In the first iteration of the derivative-free ALADIN algorithm, all x_i^* and λ^* are initialized to zero, while the numerical tolerance ϵ (stopping criterion) is set to 0.01. To evaluate the influence of key algorithmic parameters on the performance of the derivative-free ALADIN algorithm, the following parameters are systematically varied: i) matrices H_i (being multiples of the identity matrix, i.e., $0.1I$, $1I$, $10I$), ii) step-size α (0.25, 0.5, 0.75, 1.0), and iii) penalty parameter ρ (0.05, 0.5, 5, 50). The weekly scheduling problem is solved for all three scenarios S1, S2, and S3. For each scenario, all possible combinations of the above parameter values are explored. The primary performance metric is the number of iterations required for the derivative-free ALADIN algorithm to converge, i.e., satisfying the stopping criterion. In this context, the very maximum iteration limit is defined as 2000.

V. RESULTS

A. Convergence Results

The convergence behavior of the derivative-free ALADIN algorithm across the three scenarios is illustrated in Fig. 2. Each polar axis represents a fixed combination of the parameters H_i and α , with the number of iterations required for convergence plotted for varying values of ρ (indicated by the colored triangles). Cases failing to converge within the 2000 iteration limit are omitted and discussed further in Section V-B. The parameter combination achieving the fewest iterations to converge for each scenario is reported in Table II.

From Fig. 2, it can be observed that while $H_i = 1I$ yields the optimal result for scenario S1, $H_i = 10I$ demonstrates greater consistency in achieving a low number of iterations across all scenarios, particularly for S3, where $H_i = 1I$ requires significantly more iterations and fails to converge for some parameter settings. Regarding step-size α , both 0.75 and 1.0 emerge as favorable choices, with $\alpha = 1.0$ often requiring fewer iterations for convergence under the same H_i and ρ . However, $\alpha = 0.75$ appears slightly more reliable, producing the fewest iterations in two of the three scenarios (S1 and S2,

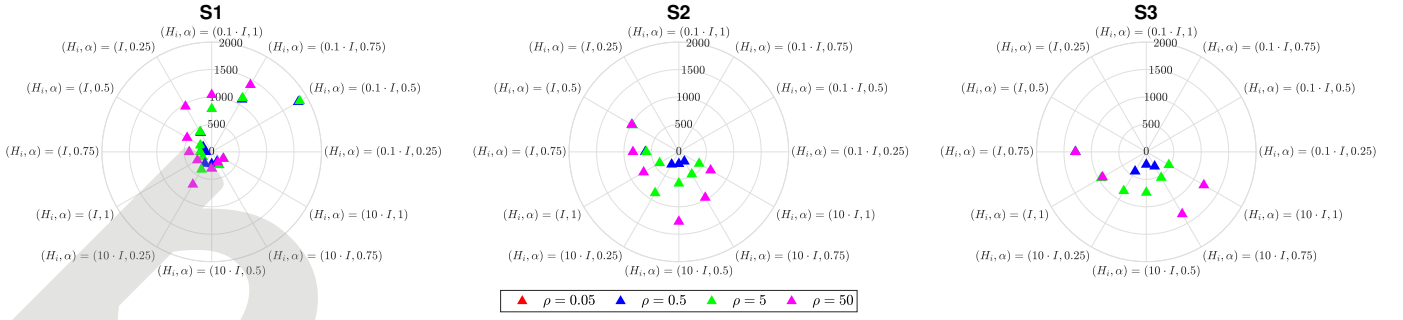


Fig. 2. Number of iterations required by the ALADIN algorithm to converge for all the three considered scenarios S1, S2, and S3, as well as for every combination of varied parameters H_i , α , and ρ .

as shown in Table II). Lastly, $\rho = 0.05$ proves to be the least effective, failing to yield convergence in any scenario, while $\rho = 0.5$ is consistently the best-performing penalty parameter.

B. Non-convergence Results

To complement the convergence analysis, the non-converging cases are classified into two categories: (i) those approaching convergence but failing within the 2000 iteration limit and (ii) those completely diverging. Fig. 3 provides an overview on these cases, presenting the percentage of non-converging ALADIN runs for each fixed parameter while varying the other two. For simplicity, $\rho = 0.05$ is excluded from Fig. 3, as it results in divergence across all combinations.

Following the conclusions from Section V-A, it becomes evident that setting $\rho = 0.5$ for fixed α , and setting $\alpha = 1.0$ for fixed ρ , minimizes the number of iterations required for convergence. However, as shown in Fig. 3, an unexpected result is that the actual combination $(\alpha, \rho) = (1.0, 0.5)$, which would intuitively be the best, leads to divergence for all values of H_i . Consequently, the optimal pair of parameters appears to be $(\alpha, \rho) = (0.75, 0.5)$, as $\alpha = 0.75$ results in fewer cases of non-convergence across all values of ρ .

This finding is important for the particular test case under investigation, because it suggests that the full-step variant ($\alpha = 1.0$), commonly recommended in the literature [8], [9], [11], may not be the best choice. Regarding the choice of H_i , Fig. 3 supports the conclusions from Section V-A, showing that $H_i = 10I$ results in a lower percentage of non-converging cases for scenarios S2 and S3 compared to $H_i = 1I$. On the other hand, $H_i = 0.1I$ performs very poorly, as none of the cases converge for S2 and S3.

TABLE II
OPTIMAL PARAMETER CONFIGURATION

scenario	number of iterations	H_i	α	ρ
S1	108	1	0.75	0.5
S2	200	10	0.75	0.5
S3	231	10	0.5	0.5

C. Solution Quality

It is important to evaluate the quality of the solution provided by the ALADIN algorithm. Accordingly, Fig. 4 illustrates the evolution of the objective function value over the iterations of the ALADIN algorithm for the optimal case ($H_i = 10I$, $\alpha = 0.5$, and $\rho = 0.5$) in scenario S3, compared to the objective function achieved by a centralized reference benchmark optimization. It can be observed that the ALADIN algorithm reaches the objective value of the centralized benchmark after a few tens of iterations. Beyond the objective function value, it is equally important to assess the solution quality from a physical perspective. To this end, Fig. 5 compares the system-level aggregator schedule obtained by the ALADIN algorithm with that of the centralized benchmark for scenario S3. The results demonstrate that the ALADIN algorithm successfully achieves the same, i.e., optimal, schedule as the centralized benchmark.

D. Computational Requirements and Scalability

The computational efficiency of the ALADIN algorithm is analyzed based on wall clock time measurements for the different steps of the algorithm. Solving individual subproblems exhibits consistent runtimes across scenarios due to their independence from the number of prosumers, with average solving times of 0.24 sec for S1, 0.28 sec for S2, and 0.28 sec for S3. However, the coupling QP emerges as the bottleneck, with its runtime increasing proportionally to the number of

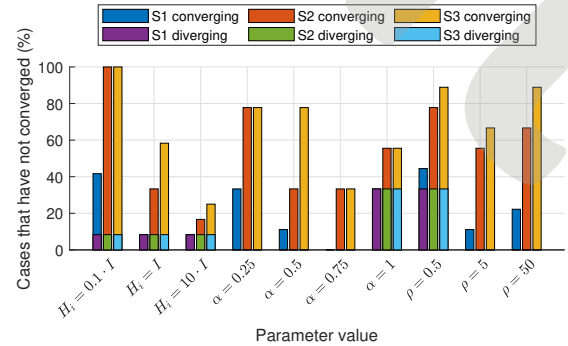


Fig. 3. Percentage of cases that have not converged for each fixed parameter.

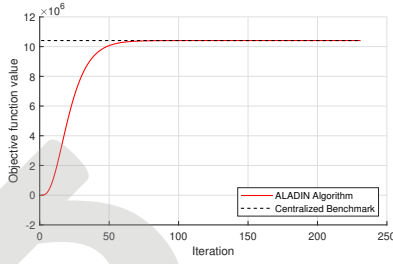


Fig. 4. Objective value of the ALADIN algorithm over iterations compared to the one of the centralized benchmark optimization for scenario S3.

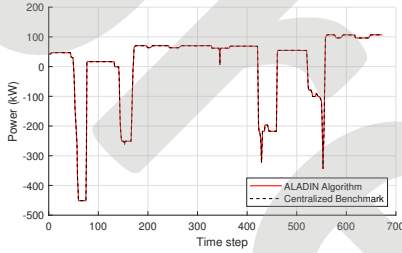


Fig. 5. Comparison of the system-level aggregator schedule obtained via ALADIN and the centralized benchmark optimization for scenario S3.

prosumers. For the optimal configuration according to Table II, the average runtime for the coupling QP is 0.93 sec for S1 (approx. 4.0 times the individual subproblem time), 4.68 sec for S2 (approx. 16.7 times), and 9.56 sec for S3 (approx. 33.6 times). Please note that these computational times are already achieved using the speed-up strategies described in Section III-A and that the regular version of ALADIN using full matrices for H_i inside the model and updating them at each iteration would result in even worse computational times for the coupling QP. This result highlights a critical challenge of ALADIN compared to other distributed optimization methods, as the increasing computational burden for the coupling QP with a growing number of subproblems puts the scalability behavior of the algorithm at risk.

VI. CONCLUSIONS

This work has studied the application of the derivative-free version of the ALADIN algorithm for weekly scheduling of aggregated energy systems considering portfolios of 10, 50, and 100 prosumers. The impact of key algorithmic parameters, including the matrices H_i , step-size α , and penalty parameter ρ on ALADIN convergence speed was thoroughly investigated.

The results demonstrated that the ALADIN algorithm is capable of providing solutions identical to those of a centralized benchmark algorithm, both in terms of the objective function value and system-level schedules, within a relatively small number of algorithm iterations. However, it was also observed that the algorithm is highly sensitive to parameter selection, where inappropriate parameter choices could significantly slow down convergence or even lead to divergence.

Future work should aim to further explore the potential of the ALADIN algorithm by explicitly benchmarking it against

state-of-the-art distributed optimization algorithms, such as ADMM. In this light, even if ALADIN may feature faster convergence in terms of number of iterations, the computational burden of solving the coupling QP problem could ultimately hinder its applicability. For instance, in the case of 100 prosumers, solving the coupling QP required computational effort that is more than one order of magnitude greater than that required for solving the individual subproblems, posing a scalability challenge for large system setups.

ACKNOWLEDGMENT

This work was supported by the German Federal Ministry for Economic Affairs and Climate Action (BMWK), promotional reference 03EWR020E (TransUrban.NRW).

REFERENCES

- [1] E. Barabino, D. Fioriti, E. Guerrazzi, I. Mariuzzo, D. Poli, M. Raugi, E. Razaee, E. Schito, and D. Thomopulos, "Energy Communities: A review on trends, energy system modelling, business models, and optimisation objectives," *Sustainable Energy, Grids and Networks*, vol. 36, 2023.
- [2] L. Moretti, E. Martelli, and G. Manzolini, "An efficient robust optimization model for the unit commitment and dispatch of multi-energy systems and microgrids," *Applied Energy*, vol. 261, 2020.
- [3] D. M. López González and J. García Rendon, "Opportunities and challenges of mainstreaming distributed energy resources towards the transition to more efficient and resilient energy markets," *Renewable and Sustainable Energy Reviews*, vol. 157, 2022.
- [4] S. Schwarz and A. Monti, "Computational Performance Study on the Alternating Direction Method of Multipliers Algorithm for a Demand Response Peak Shaving Application," *IEEE Systems Journal*, vol. 17, no. 2, 2023.
- [5] N. Gatsis and G. B. Giannakis, "Residential load control: Distributed scheduling and convergence with lost AMI messages," *IEEE Transactions on Smart Grid*, vol. 3, no. 2, 2012.
- [6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. and Trends in Mach. Learn.*, vol. 3, no. 1, 2010.
- [7] M. F. Anjos, A. Lodi, and M. Tanneau, "A decentralized framework for the optimal coordination of distributed energy resources," *IEEE Transactions on Power Systems*, vol. 34, no. 1, 2019.
- [8] B. Houska, J. Frasc, and M. Diehl, "An augmented Lagrangian based algorithm for distributed nonconvex optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, 2016.
- [9] A. Engelmann, Y. Jiang, T. Muhlperfordt, B. Houska, and T. Faulwasser, "Toward distributed OPF using ALADIN," *IEEE Transactions on Power Systems*, vol. 34, no. 1, 2019.
- [10] A. Murray, T. Faulwasser, and V. Hagenmeyer, "Mixed-Integer vs. Real-Valued Formulations of Battery Scheduling Problems," in *IFAC-PapersOnLine*, vol. 51, no. 28, 2018.
- [11] A. Engelmann, Y. Jiang, H. Benner, R. Ou, B. Houska, and T. Faulwasser, "ALADIN- α —An open-source MATLAB toolbox for distributed non-convex optimization," *Optimal Control Applications and Methods*, vol. 43, no. 1, 2022.
- [12] B. Houska and Y. Jiang, "Distributed Optimization and Control with ALADIN," in *Lecture Notes in Control and Inf. Sci.*, 2021, vol. 485.
- [13] 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.
- [14] Institut Wohnen und Umwelt GmbH and the Consortium of EU IEE Project EPISCOPE, "National building typologies." [Online]. Available: <https://episcopes.eu/building-typology/>
- [15] Bundesverband der Energie- und Wasserwirtschaft e.V., "German standard load profiles." [Online]. Available: <https://www.bdew.de/energie/standardlastprofile-strom/>
- [16] Deutscher Wetterdienst (DWD), "Open data server of the German meteorological service." [Online]. Available: <https://opendata.dwd.de/>
- [17] Gurobi Optimization LLC, "Gurobi 12.0 Optimizer Reference Manual," 2024. [Online]. Available: <https://www.gurobi.com/>