Energy Informatics

RESEARCH

Open Access

MASSIVE: A scalable framework for agent-based scheduling of micro-grids using market mechanisms

Jakob M. Fritz[1*], Lea Riebesel[1], André Xhonneux[1] and Dirk Müller[1,2]

*Correspondence:
Jakob M. Fritz
j.fritz@fz-juelich.de
[1]ICE-1, Forschungszentrum Jülich, Wilhelm-Johnen-Straße, Jülich 52428, Germany
[2]E.ON ERC, RWTH Aachen, Mathieustraße 10, Aachen 52074, Germany

Abstract
With the increasing share of distributed renewable energy sources the need arises to store excess energy and/or to shift demands to match the given supply. To coordinate multiple suppliers and demands in a local energy-system different control approaches can be used. This publication introduces a framework called MASSIVE that aims to coordinate multiple participants in a district energy-system. The energy-system is controlled in a distributed way by using a multiagent approach that is scheduled by a market-mechanism. This market-mechanism allows to coordinate many individual agents with only few restrictions by using pricing mechanisms. This offers an incentive for the agents to adapt their power consumption to best match the forecasted power supply. However, the agents are free to follow this incentive or ignore it depending on the value of the incentive. The individual agents are flexible in the internal approach to forecast power supply or demand, allowing easy development of agents using individual algorithms. The coordination takes place using a market-mechanism that is similar to the day-ahead market. It, however, is run multiple times a day to form a rolling horizon, making it less sensitive to forecasting errors. The market approach furthermore exhibits a nearly linear scalability with regard to the duration of the market clearing. On the used computer, the creation and solving of the linear optimization-problem is performed in less than one minute for approximately 1500 participating agents. Therefore, this approach is capable of real-time use and can be used in real-world applications.

Keywords Agent-based scheduling, District energy-systems, Micro-grids, Market-based

Introduction

To match demand and supply in electrical energy grids multiple ways can be used. Adjusting the supply-side can be complicated (depending on the type of power plant) or strongly limited, as renewable energy sources can be reduced in power output but not increased. However, reducing the output of renewable energy power plants means, that renewable energy supply remains unused in order to not overload the power grid. Therefore, it gets more relevant how to adapt the demand to match the supply best. One way

© The Author(s) 2025. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

to do so is to add storages. Another way is, to adapt the demand of the existing consumers so that they can shift their electrical demand over time, called Demand Side Management (DSM). An example of those shifted demands is the usage of heat-pumps, that can heat up houses or hot water storages during times of cheap (and excess) electricity and reduce their consumption in times of little renewable power supply.

In order for this shift to work, communication between the consumer and the producers is needed. Additionally, the consumer should also communicate with other consumers to prevent situations, where multiple consumers shift in the same way, also leading to an overshooting of the system. Furthermore, a better matching of supply and demand can be done in local micro-grids, the less the overall distribution grid is stressed.

This publication presents a modular, open-source framework, that can be used to coordinate local micro-grids for districts, campuses, and the combination of multiple energy systems in general. It uses time-discretization and repeatedly performs power dispatch, forming a rolling horizon. In this power dispatch, not only the next time-step is taken into account but the whole upcoming horizon (in the examples the next 24 h). By including the whole horizon in the market mechanism, the benefit of load shifting can be directly estimated. As the action in one time-steps influences all later time-steps, the time-steps in the horizon cannot be solved independently, but are coupled with each other. This inclusion of many coupled time-steps (the whole horizon) in a market-mechanism for the application in agent-based scheduling of micro-grids has not been published before, to the knowledge of the authors.

Due to its modularity, the presented framework can help analyze the influence of individual components, as they can easily be replaced by different implementations.

The paper is structured as follows: First, some background to used terminology and some commonly used approaches is given. Chapters 3 (Main aims of MASSIVE and differences to existing approaches) and 4 (Structure of MASSIVE) present the main goals and the structure of the newly developed distributed control ("MASSIVE"). Chapter 5 (Scalability for district systems) indicates how the overall system scales followed by chapter 6 (Additional modes of operation) that introduces more modes of operation. Chapter 7 (Conclusion and outlook) concludes this publication and gives ideas for further development.

Theoretical background

To highlight the new aspects of MASSIVE, first some terminology is specified. This includes used timescales, as well as differences between centralized, decentralized and distributed control approaches.

Differentiation between control, scheduling & planning

When discussing how to run a plant (or system in general) multiple terms occur. In order to avoid misunderstandings, the terms "control", "scheduling" and "planning" shall be defined and explained in this section. The definition of these terms is adapted from [1–3]. One of the main differences is the time horizon of each of those decision-making processes. A rough estimate of the length of horizons is given in Figure 1.

To avoid confusion, the term of "optimization" will not be used to refer to some decision-making process, but only to the approach of solving mathematical problems.

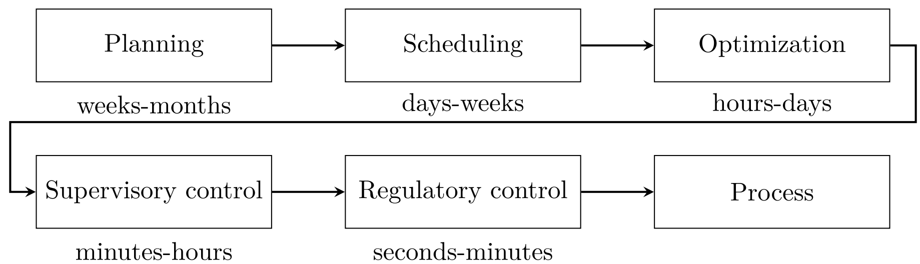

Fig. 1 Hierarchy of planning, scheduling and control and used time-steps and horizons for each of them. Adapted from [2]

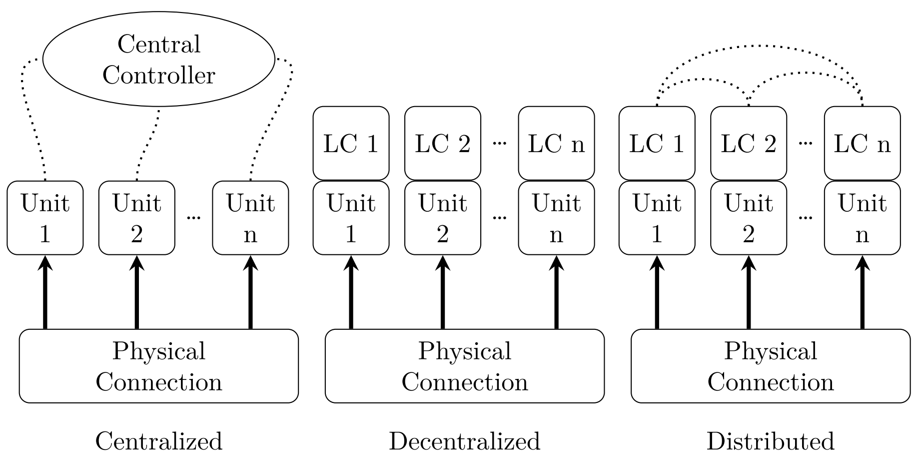

Fig. 2 Comparison of control schemes *Unit* represents a local consumer or producer and *LC* a Local Controller. In a distributed control scheme, the local controllers can either be connected to all other LCs or only to some others. The "physical connection" links units together by physical processes as e.g. heat transfer or because they are coupled to the same power-grid, i.e. they influence each other. It does not imply any communication. Communication is depicted as dotted lines in the upper part of the figure. Based on [5]

The approach presented here deals with a resolution of a few minutes (currently time-steps with a duration of 15 minutes are chosen) and a horizon of one day. This horizon also may be extended if necessary. With these durations, the approaches presented in this work can be classified into the field of supervisory control up to scheduling (when avoiding the term "optimization" as mentioned above). For the rest of the publication, the term scheduling will be used.

Centralized, decentralized and distributed approaches

Control systems can be categorized into centralized, decentralized and distributed systems according to [4]. A visualization based on [5] is shown in Figure 2.

In a centralized control, there is only one single instance doing the control, and other parts are only providing data and taking care of implementing decided actions. Whereas in a decentralized control, there are many instances, that each control their part subsystem, but do not communicate to coordinate themselves with each other. In a distributed approach, the individual instances, each control a subsystem, but communicate with each other [4].

For MASSIVE a distributed control approach is chosen as it combines aspects, that are relevant for agent-based control. Each local controller knows more about the state of the whole system than it would do in a decentralized control. This leads to better decisions and therefore getting closer to the overall optimum. Another advantage is the increased resilience to communication disturbances compared to a centralized control. This is especially true for systems without single points of failure.

Distributed multiagent systems

Among other ways, distributed control can be implemented as multiagent control. In a multiagent approach, the single entities (autonomous elements of the energy system) are seen as agents (cf. [6]). Agents can have the ability to control their subsystem and react to unforeseen changes in their subsystem in a more advanced way than a simple consensus-based controller. Further information on such multiagent approaches can be found e.g. in [7, 8]. The individual agents can have their own (and different) control mechanism. This can be a rule-based approach, but also an analytical or optimization-based approach. In this publication, agents with rule-based control, as well as with model predictive control are used. More information of the combination of model predictive control and distributed control can be found in [9].

Each agent adapts its behavior depending on the state of the system. Since the state of the system is dependent on the behavior of each agent, each agent adapts its behavior indirectly depending on the decision of the other agents. Therefore, the order in which the agents chose a decision may be of relevance. Hence, the coordination of the agents can also be differentiated depending on when the local controllers are updated. In general, there are three possibilities on when to update the single agents: One after the other (serial), all in parallel, or something in between. A general comparison on different communication schemes can be found in [10, 11].

Main aims of MASSIVE and differences to existing approaches

The framework introduced here is called MASSIVE, short for "MultiAgent Scheduling Solution in a Virtual Environment". The development of MASSIVE follows the aims, that are described and reasoned below.

As the name already states, MASSIVE is a framework for an agent-based control approach. This concept is chosen, as it enables easier development and maintenance as well as simpler addition and removal of individual parts compared to a centralized control system. Agents can represent different types of participants. Those can be consumers, producers or prosumers. They all can follow a similar internal structure, leading to simple adaptation if the real-world system changes. This general structure for agents shall make it easy to participate in this system from a technical side. To reduce the entry hurdle even more, exemplary implementations for different agents are released open-source to provide starting points for the own individual systems. The power consumption or power supply of the agents is communicated for discretized time-steps of a fixed length. Within this time-step the power is assumed to be constant.

The coordination of the agents is done by a central agent called "central instance". This instance communicates with all agents and does the power dispatch by using market clearing. The market clearing is performed repeatedly for the upcoming time-steps, forming a rolling horizon. The market clearing is inspired by the day-ahead market clearing used at the European Power Exchange EPEX Spot [22]. However, the market of MASSIVE is cleared much more frequently, so that subsequent market clearings also have a strong overlap in the time-steps they take into account. Another difference between the day-ahead market and the market-mechanism in MASSIVE is the different structure of bids, allowing for easier participation in MASSIVE. More details on the used structure of bids is given in subsection Marketplace and market-mechanism.

Compared to a central approach, this offers the advantage of being more privacy-friendly, as fewer data needs to be shared with other entities. Furthermore, it is easier to parallelize as the individual agents are only coupled via the repeated market clearing. More general comparisons of central and decentralized approaches have already been published (e.g. in [12, 13]).

The distributed approach presented here offers a higher modularity than a centralized approach, where all information needs to be transferred to a single entity and are processed by a single approach (be it rule-based, optimization-based or otherwise). This distributed approach offers the combination, as different agents can follow different methods to decide their own forecast. Additionally, the communication is based on a commonly used protocol. This conceptually enables the use of different programming languages within agents, which is not possible in a central control.

Within the distributed approaches, some work without a central instance (e.g. [14–16]). There, the agents (sometimes also called *participants*) often communicate with their direct neighbors, but sometimes also with the whole network. In contrast to this, MASSIVE uses a central instance to provide a single point of communication. Therefore, the agents do not need to communicate with multiple partners, but only with a single one. Furthermore, the agents do not need to know which other agents are part of the network, improving privacy of the approach.

Some distributed approaches (e.g. [16–18]) also use a central instance, but often handle the agents in a sequential process. In MASSIVE the agents can calculate their new forecast in parallel, making the approach more scalable to higher number of agents while staying applicable to real-time applications. Another approach also updating agents in parallel (and in an iterative fashion) is [19]. However, this approach does not take coupling of time-steps into account, but solves the separate time-steps independently. The approach there also allows only parts of the agents to update their bids. So only the sellers or buyers are allowed to update, depending on if there is more supply or demand. By this, no optimization is needed, but the market clearing can be performed rule-based.

In addition, MASSIVE can be used to compare the multiple ways of coordination, by supporting single-shot and iterative paths (cf. section 7). The single-shot approach performs a single market clearing for each starting time, whereas the iterative approach performs multiple market clearings in order to converge towards a solution per starting time. The latter is computationally more expensive but can be beneficial if the system at hand is highly volatile (e.g. the forecasts deviate much between consecutive starting times). This support of multiple approaches is not done by another solution to the knowledge of the authors.

As a last differentiation to many other solutions, MASSIVE clears the market for a complete horizon of time-steps, as opposed to other solutions, that clear only the next time-step without taking further steps into account (e.g. [20, 21]). Therefore, those solutions do not do scheduling by the definition above, and do not encourage shifting of demands across time-steps. Also approaches, that take the whole horizon into account do not couple time-steps, so they clear the market for the time-steps independently (e.g. [19]). This makes it harder for participating agents to perform the Demand Side

Management mentioned above, as no restrictions between time-steps are taken into account (e.g. inertia of units or the need for consecutive demand).

This combination of a distributed, agent-based control employing a central market mechanism to clear a rolling horizon and allows for a parallel update of participating agents is not found in literature to the knowledge of the authors.

Furthermore, as the source code of MASSIVE is publicly available and MASSIVE follows a modular structure, it can serve as a basis for own examinations and comparisons.

Structure of MASSIVE

After introduction of the main lines of the concept, the following section will introduce the structure of the interaction of the components and will also show how the agents themselves are structured. The shown agents are used for the example case below but can easily be replaced or adapted to fit the need for the specific situation.

This section will start with an overview of the interaction between the agents, followed by an introduction into the used market-mechanism and the different types of agents employed here. These can be split into two groups. Agents that perform optimization in order to determine the needed or supplied power and agents that perform rule based decisions of the power, they can provide or need.

Overall structure

As can be seen in Figure 3, the whole system is split into multiple agents.

Those communicate via a specified protocol. In the current version, MQTT is used, as this is widely available on many platforms, an established standard and can deal with many messages in brief time-scales. Since communication is done via network connections, it is trivial to distribute the agents to multiple machines, leading to an easy deployment in across buildings.

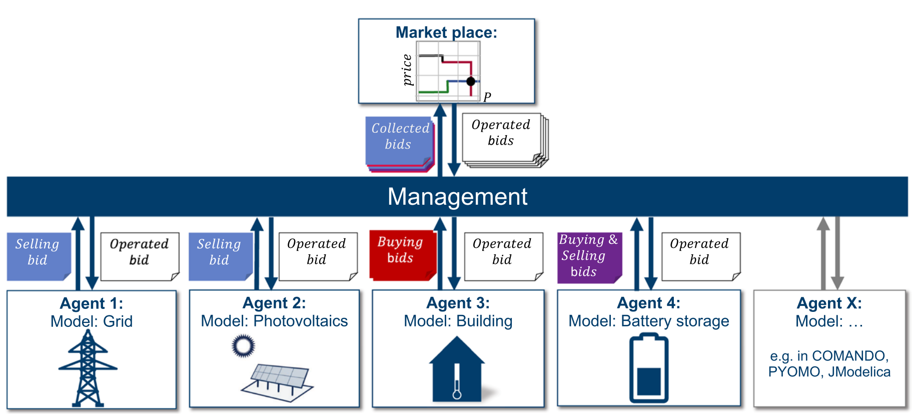

Fig. 3 The overall structure of the MASSIVE-Framework. It consists of multiple agents, an instance for coordination and an instance for central decision-making. This decision-making is done in form of a marketplace. For this marketplace each agent hands in one or multiple bids (buying and or selling bids) and it receives information which bid got accepted and will be turned into practice. More details to each of the building blocks can be found in the following sections

Each agent is allowed and requested to hand in one or multiple exclusive bids (i.e. multiple bids cannot be combined). Each bid specifies a power forecast and a price for that power. Both, the power and the price are time-dependent for multiple time-steps of the horizon. In the examples shown in this publication, the horizon is always 96 time steps of 15 minutes each. Therefore, the power and price do not need to be constant over the upcoming forecast. The power forecast of each time-step can be a demand or a supply of power, depending on the sign of value handed in (positive for supply and negative for demand). Depending on the sign of the power forecast, the communicated price for that time-step is automatically interpreted as upper (for demand) or lower (for supply) price limit.

The communication between the market mechanism and the individual agents is organized by a management layer. This layer is responsible for requesting and collecting the bids as well as for communicating the results of the market clearing. That approach increases the modularity and makes it easier to adapt the market mechanisms if wanted.

The scheduling here works on time-steps of 15 minutes. Therefore, aspects like stability of the power frequency are outside the scope of this approach, as is also in other market mechanisms. To deal with stability of power frequency, other means of multi-level reactive power are needed. This is also similarly done in the day-ahead market at the European Power Exchange (EPEX) [22], where the aspect of frequency stability is not regarded.

As shown on the right, more agents could be included in the system, than presented here. Those agents can be added as needed. This also includes agents with different internal mechanisms as well as different programming languages used, as the agents only interact via JSON-encoded data in MQTT-messages. So not only MPC-agents using COMANDO [23] or PYOMO [24, 25] are possible, but also using e.g. JModelica [26] or other frameworks. For the sake of simplicity, this paper only uses agents depicted in the figure.

In the examples shown in this publication, it is assumed, that all agents are owned and operated by the same entity. However, this does not need to be true. If the agents are owned and operated by different entities, this approach shows its advantage of the increased privacy compared to other approaches (e.g. centralized or distributed ones with more specific communication.) The distributed ownership only needs to agree on the use of the central market-mechanism. As the whole source code is publicly available, all involved parties have the option to check the correct operation of the market mechanism. The cost for operating the server to run the market clearing and the communication need to be shared by the participating parties.

If the agents are owned by different parties it needs to be emphasized, that MASSIVE does not include any form of payment mechanism. This needs to be done outside MASSIVE with the uniform power price and the traded volumes as determined by MASSIVE.

Communication between agents and marketplace

As the management coordinates the whole communication, it queries all agents to provide bids for the next market clearing. The query contains information about the upcoming horizon such as the number of time-steps and the duration. It furthermore includes the prices that were determined in the previous market clearing to provide some guidance.

The bids by the agents can be selling bids, buying bids, or combined bids. On a technical level, the agents do not need to make the distinction between the types of bids, as this is implicitly clear by the value of transmitted power.

Each bid contains at least one variant how the agent forecasts to consume or supply power. Optionally, the agents can hand in multiple variants. This is helpful, if the systems can be operated in different ways, as is the case for some heating-systems and energy storages. If the agent hands in multiple variants the market-mechanism can decide which one to choose, but at most one variant can be implemented. Therefore, the agent does not need to take combinations of variants into account.

A single variant mainly consists of two lists of values for the upcoming horizon. One list for the amount of power the agent plans to supply or consume and the second list for the price the agent is willing to accept for this power. The sign of the values for the amount of power implicitly states if the agent plans to supply power (positive value) or to consume power (negative value). Not only the values can differ for each time-step in the forecast, but also the sign does not need to be the same for all time-steps. Therefore, the same agent can act as supplier in one time-step while consuming power in another time-step. The price can also be specified for each time-step separately, and the value always denotes the limit. Depending on whether the agents want to sell or to buy power (in that time-step), the price is the lowest or highest acceptable, respectively.

There are more optional pieces of information, the agent can communicate. For example, it can be specified, whether the communicated amount of power is only an upper limit or if it shall be dealt with in an "all or nothing" manner.

If agents fail to reply within a specified time-frame, the management assumes the last forecast is still valid, when shifted by one time-step. With this assumption, the agents do not drop out completely, and most of the time this assumption is more correct than assuming the respective agent will neither supply nor consume any power. This consequence for missing replies is used for all approaches and all examples in the following sections.

Because of this approach, the distributed approach is more resilient to communication-issues than a centralized approach. There, more data needs to be transferred (all sensor-data compared to the bids) and an outage impacts the system more, as the decision-making is done centrally but could not be communicated. In this distributed approach, the agents can decide on information from the previous time-step, that surely are outdated, but still more realistic than a predefined default value.

As an additional safeguard, MQTT has a function to tell the management if an MQTT-participant left unexpectedly. If this is the case, the respective agent is assumed to be gone (so neither supplying nor consuming power) and is removed from the market-mechanism completely.

The operation instructions (operation bids) by the marketplace follow a similar form as the bids by the agents but contain only a single list of values for power for each agent. They furthermore include the determined prices for all upcoming time-steps. The price is the same for all agents (called uniform pricing).

Marketplace and market-mechanism

The main part of coordinating the supplied power is done by means of a market-mechanism. After the market clearing it is decided which agent is supposed to provide how much power for the upcoming time-horizon.

The structure of the market-mechanism itself is similar to the market-mechanism of a day-ahead market. First, all agents can hand in their bids. After this, the market is cleared by solving an optimization problem formulated to be a mixed integer linear problem (MILP). The creation and handling of the problem is done by PYOMO [24, 25]. The formulation as MILP leads to fast solving times even with many participants (see Section Scalability for district systems for more information). The main objective of the market (Equation (1)) is to maximize social welfare. The mathematical formulation of the objective and the constraints can be seen below. Social welfare is defined as the summed price buyers are willing to spend, subtracted by the summed price at which sellers offer the goods (cf. [27]). Visually speaking, social welfare depicts the area between the two curves of a market (quantity of supply and demand vs. price per unit). In [27] more information and examples of social welfare in the context of electricity markets can be found, while [28] provides some more general information on the concept and definition of social welfare. With social welfare, each agent can specify the individual costs it has to provide power or the cost it is willing to pay for consumption. With this approach, the cost for operating the general infrastructure is not covered. Those costs for operation of the market and communication as well as the transmission of power need to be calculated and charged separately.

Although the formulation is similar to the day-ahead market, the market used here is run not only once a day, but regularly after each time-step, so that a rolling-horizon forms. In the examples shown in this publication, the time-steps span 15 minutes each and the horizon consists of 96 time-steps, leading to a total horizon of 24h. Only the first time-step is the applied, while the other time-steps of the horizon are only used to decide on the strategy for the upcoming time. In contrast to the day-ahead market at EPEX [22], no block-bids are allowed and all bids of each agent are treated as exclusive bids. Both decisions lower the entry barrier and make it easier to participate in the proposed market clearing.

Parameters are written in uppercase (e.g. $P_{\min}, P_{\max}$), whereas variables are written in lower-case letters (e.g. p_o, p_{used}). The parameters for minimal and maximal power, that an agent can provide or consume are $P_{\min}, P_{\max}$. The price, that is each agent defines to sell or buy power in a specific option and a specific time-step is called $PR(a, o, t)$. The power, that is proposed in a specific option is denoted by p_o and the power that shall be used by an agent is called p_{used}.

The only integer variables are whether to use an option of an agent or to discard it. These variables are $acc(a, o) \in \{0, 1\}$ and $acc_1(a, o) \in \{0, 1\}$ for all agents and all options of those agents individually. The variable $acc(a, o)$ determines if the option is used for all or for no time-steps. However, if the option is discarded, it is discarded for all time-steps. The variable $acc_1(a, o)$ determines if the agent's option is used for the first time-step. The two binary variables are coupled via constraint 7.

T is the set of time-steps t of the whole horizon. A is the set of used agents (individually denoted as *a*). O_a is the set of options (denoted as *o*) provided by agent *a*.

$$\text{minimize} \sum_{t \in T} \sum_{a \in A} \sum_{o \in O_a} p_o(a, o, t) \cdot PR(a, o, t) \tag{1}$$

$$\text{s.t.} \qquad p_o(a, o, t) \geq P_{\min}(a, o, t) \cdot acc(a, o) \qquad \forall a \in A, o \in O_a, t \in T \tag{2}$$

$$p_o(a, o, t) \leq P_{\max}(a, o, t) \cdot acc(a, o) \qquad \forall a \in A, o \in O_a, t \in T \tag{3}$$

$$p_{used}(a, t) = \sum_{o \in O_a} p_o(a, o, t) \qquad \forall a \in A, o \in O_a, t \in T \tag{4}$$

$$\sum_{o \in O_a} acc(a, o) \leq 1 \qquad \forall a \in A, o \in O_a \tag{5}$$

$$\sum_{a \in A} p_{used}(a, t) = 0 \qquad \forall t \in T \tag{6}$$

$$acc_1(a, o) \leq acc(a, o) \qquad \forall a \in A, o \in O_a \tag{7}$$

$$p_o(a, o, 0) \geq P_{\min}(a, o, 0) \cdot acc_1(a, o) \qquad \forall a \in A, o \in O_a, t \in T \tag{8}$$

$$p_o(a, o, 0) \leq P_{\max}(a, o, 0) \cdot acc_1(a, o) \qquad \forall a \in A, o \in O_a, t \in T \tag{9}$$

$$\begin{aligned} & acc_1(a, o) \cdot acc_1(a2, o2) \cdot PR(a, o, 0) \leq acc_1(a, o) \cdot acc_1(a2, o2) \cdot PR(a2, o2, 0) \\ & \forall a, a2 \in A, \\ & o \in O_a \mid P_{\min}(a, o, 0) > 0, \\ & o2 \in O_{a2} \mid P_{\max}(a2, o2, 0) < 0 \end{aligned} \tag{10}$$

The first constraints (Equations (2) and (3)) ensure, that the power is within the range specified by the corresponding bid of the agent. Constraint 4 simplifies further calculation as it couples the power that is demanded or supplied by an agent to the options provided by that agent. Equation (5) ensures, that at most a single bid is followed, since the bids are exclusive and cannot be combined. This decision was made to simplify the creation of bids by the agents. If bids could be combined, it would be harder for agents to hand in multiple bids, as they would need to check if all bids could be combined. By the design-decision to use exclusive bids, this issue is avoided.

Equation (6) constraints the optimization to match supply and demand for all time-steps.

Constraints (7) to (10) are relevant only for the first time-step. In all later time-steps the objective leads to the behavior of only using suppliers if they provide power cheaper than the consumers are willing to pay. This, however, only holds true for the sum over the whole horizon and not necessarily for all time-steps individually. As the first time-step is executed, it is important to only use suppliers that are cheaper than the lowest accepted price of consumers. For this, additional binary variables are used that indicate if an option of an agent is used in the first time-step (Equation (7)) and constraints the power of that agent accordingly (Equation (8) and (9)). The last Equation (10) makes sure that two options can only be used if the supplier is cheaper that the demand. If any of the variables $acc_1(a, o)$ or $acc_1(a2, o2)$ becomes 0, the constraint becomes trivial. This constraint is applied only to combinations of options, where option (a, o) is able to provide power (i.e. $P_{\min}(a, o, 0) > 0$) and the second option (a2, o2) is able to consume power (i.e. $P_{\max}(a2, o2, 0) < 0$).

Exemplary MPC agents

This section will introduce and explain the agents employing model predictive control. Those agents represent the behavior of a heating in a simplified building or the scheduling of a battery storage. Because of the modularity of MASSIVE, different agents are possible to integrate. However, only two are shown here, as they are used in the presented exemplary cases afterwards. The agents contain a model of the unit or building they control. This model is used in a model predictive control to decide the optimal schedule for power consumption (or supply).

Using MPC enables these agents to adequately adapt to changing external parameters as well as to uncertainties of those parameters. The presented MPC agents here differ in the range of their possible behavior. In the context of the use-case given below, the MPC based approach is used in two agents. One represents the heating of a generic building, the other one a battery energy storage. Agents with other models are possible, but not used in this example. The heating agent is able to shift demand over time to reduce or avoid consuming energy during periods of expensive power. The battery agent however is not only able to reduce its demand, but it can also supply power to the market and therefore to other agents.

Heating agent

The heating agent represents the heating of a simplified building. This is modeled as a large room with a wall connecting the inner air volume to the ambient temperature. Furthermore, the model includes a concrete core that can be heated up to store thermal energy that is transmitted to the air volume over time. The model is described in more detail in [29] and is formulated as an MILP.

This simplified model can be parameterized to vary the size of the room as well as how well the room is isolated. Therefore, multiple buildings can be simulated behaving differently. As the model is formulated as MILP, it can be optimized with many available solvers. Therefore, the optimization shifts power usage to times with lower power price to reduce overall cost, if this is in accordance with constraints of the model representing comfort of people. Because the walls transmit heat from the internal air volume to the outside ambient air, storage of energy is only feasible for limited amounts of time, as the room and the concrete core lose heat. This is included in the optimization problem and implicitly limits the time that shifting the load for heating still is feasible.

Battery agent

The battery agent is used to store energy and provide it at a later point acting as consumer in some time-steps and as supplier in others. The model is formulated as MILP and includes financial penalties for aging effects. Calendaric as well as cyclic aging are taken into account in the MILP formulation.

Including these agents illustrates how prosumers can behave in such a situation, where a market is repeatedly cleared. It furthermore adds volatility to the supply side, as the supply side is now influenced by the power price even more. This helps to show how market clearing works when both sides are elastic, as opposed to markets where only demand depends on the price (in case of power sources that have very small reduced costs, as it is e.g. the case for renewable power sources) or where only the supply depends on the price, as currently, most households have a static electricity price and therefore no incentive to change their demand depending on the availability of cheap electricity.

The penalties within the battery model include two different mechanisms. One is a mechanism representing the calendar aging of a battery, the other is the cyclic aging of the battery. The calendar aging is not dependent on the usage of the storage whereas the cyclic aging only depends on the usage. For the calendar aging, the cost for renewal are divided by the duration a battery takes on average to lose 20% of capacity. From this calculation an average cost per time-step can be derived and taken into account in the optimization and when determining if and how much profit the battery can achieve. As simplification for the cyclic aging, the costs for replacing the battery are divided by the number of cycles. Therefore, these costs are averaged of the whole period of usage. This is done as otherwise (with a decreasing cost of cycles) the first cycles would be inhibitory expensive to ever be used. These two aging mechanisms are then added up to represent the total degradation.

With these two simplified aging mechanisms, the battery model can remain piecewise linear and therefore much easier to solve compared to a mixed integer nonlinear model (MINLP).

Exemplary analytical agents

Apart from agents using optimizations to decide on their behavior, also simpler analytical calculations can be used. The agents shown here are not able to shift their demand or supply. Because of this, no optimization is needed. The two agents described in the following use a physical model to determine the possible power provided by PV-modules. Since this can be derived analytically, no optimization is needed. The second agent, that represents the connection to the distribution-grid also does not need to use optimization to generate a forecast, as the distribution-grid is large compared to the microgrid under examination and therefore can supply enough power at all times.

PV agent

An example for agents that derive their power supply by analytical equations is the agents representing a photovoltaic installation. This agent uses weather-information to estimate the upcoming electrical power it will be able to provide. To provide a more general information about supplied PV power, the model is not used with current weather-forecasts, but with data from average years (Test Meteorological Years; TMO) instead. These TMO represent an average year in a specified location. They are used to provide a more generalizable example compared to using weather data from a single specific period of time. This does not only include data of clouds, but also ambient temperatures, as the efficiency of PV panels is temperature dependent. As the forecast can be updated every time-step, a real life application would include forecasts that are more precise for the nearer future compared to time-steps in the more distant future. As this simulation uses TMO data the forecast always equals the actual weather. Therefore, the forecast can be seen as precise for all time-steps (perfect forecast). If in reality the actual power supply of the PV agent deviates from the forecast (within the current time-frame of 15 minutes), the difference could be compensated from the power grid agent (see below), as the microgrid is not an island system. If the supply is expected to deviate for more than 15 minutes (or the duration of a time-step in general), the forecast can and should be updated, so that the market mechanism can take this change into account in the next market clearing.

The PV agent models the electricity production from PV panels that have a fixed orientation. A single axis or even two-axis tracking could also be used, but is rarely found in Germany. Thus, a fixed mount is used in the model. The orientation of the PV installation is set as parameter and can be varied across multiple instances of the agent to avoid identical agents.

In the following examples, multiple PV agents will be included. Those agents will have a different parameterization, as this is more representative of a real-world application.

The used model of the photovoltaic installation is provided by the *pvlib* package [30] in version 0.9 [31].

Power grid agent

Another agent providing energy represents the connection to the local distribution grid. This agent is assumed to provide large enough amounts of power to satisfy the demand of the local consumers. Furthermore, it is assumed to have no inertia for providing power. This is a reasonable assumption, as the power grid is able to react on smaller time-scales compared to the used 15-minute intervals used in this micro-grid.

The price of the provided power is assumed to be constant over time. This assumption is chosen as most contracts contain a fixed price that does not depend on the energy-price at the larger power exchanges. However, the used code is also capable of dealing with prices varying for different time-steps.

Assumptions made for the energy-system

After briefly introducing the used agents, there are also assumptions that are made for the energy-system itself.

As already stated with the agent representing the connection to the local distribution grid, this micro-grid is assumed to be connected to the outside all the time. This also indicates the scope of this system. It is assumed that frequency regulation is done from the outside. The scheduling-system presented in this publication does not take care on frequency regulation. It also does not take care on regulatory power for timescales below 15 minutes. As the connected distribution grid is much larger than the controlled micro-grid, it is assumed, that the distribution grid can provide enough power to compensate for smaller deviations in power supply and demand. The forecasts of the individual agents are (as all forecasts) to some extent incorrect, leading to a non-perfect scheduling of the whole system. Also, MASSIVE schedules power consumption and supply for the upcoming time-steps, but it cannot intervene if agents do not stick to their own forecast. This behavior could be penalized financially to make it unattractive for agents to deviate from their own forecast.

Scalability for district systems

After the general concept is shown, it is also important to study how well the system scales if it is used with many participating agents.

General remarks on scalability

Since the agents themselves are separate and do not communicate with each other they are easy to parallelize. Therefore, they should not pose a problem with respect to computational power, when increasing their number. The used communication-framework MQTT is also able to deal with large amounts of messages in very short time. Therefore, the main focus will be on the scalability of the marketplace, as this is the central building block and cannot be parallelized easily. The marketplace is technically an optimization problem whose size is directly dependent on the number of agents participating in the market-mechanism.

Scalability of the market

To examine the scalability of the market-mechanism, multiple scenarios are run that differ in the number of participating agents. The number of different agents is increased at the same rate to keep the ratio between adaptive demands (i.e. heating agents) and renewable energy sources (i.e. PV) constantly at one PV agent per heating agent, while there is always only a single power grid agent (regardless of the number of PV and heating agents). Furthermore, the agents are set to exhibit a slight randomization in parameterization. This avoids having many identical agents with identical profiles and therefore increases realism of the examination. This randomization is applied to the scaling of the building in the heating-agent, as well as the orientation of the PV.

For the PV agents, the orientation follows a normal distribution centered around a south-orientation with a standard deviation of 45°. For the heating agents, the enclosed air volume follows a normal distribution centered around 195 m^3 with a standard deviation of 10%. Both randomized parameters are capped at $\pm 2\sigma$ to avoid orientation of PV panels towards north and to keep the size of buildings roughly comparable.

The randomization is initialized with a fixed seed at startup. This ensures, that multiple runs of the simulation produce the same results, while still creating agents that are different from each other.

Multiple durations are taken into account and displayed in Figure 4. There, the averaged durations are shown as well as polynomials of the form $a + b \cdot x^c$ that are fitted to all durations instead of only the averages shown. The idea behind the fits is to guide the eye and to examine the scaling behavior.

"Model creation duration" is the duration it takes for PYOMO to create the model/problem formulation from the bids of the agents. "Market clearing duration" is the wall-clock time between transferring the problem to the solver and receiving the result. The market clearing duration scales higher than linear with an exponent of 1.46 ± 0.18. When summing up the durations for model creation and for market clearing the sum scales still higher than linear, but only with an exponent of 1.16 ± 0.12 Nevertheless, the duration for many agents (e.g. 500 building agents, leading to ~1500 agents in total) the total duration of model creation and solving is below one minute. Therefore, even larger optimizations within the individual agents are possible, as the central coordination is rather fast. The optimization is performed using the HIGHS solver via the Python

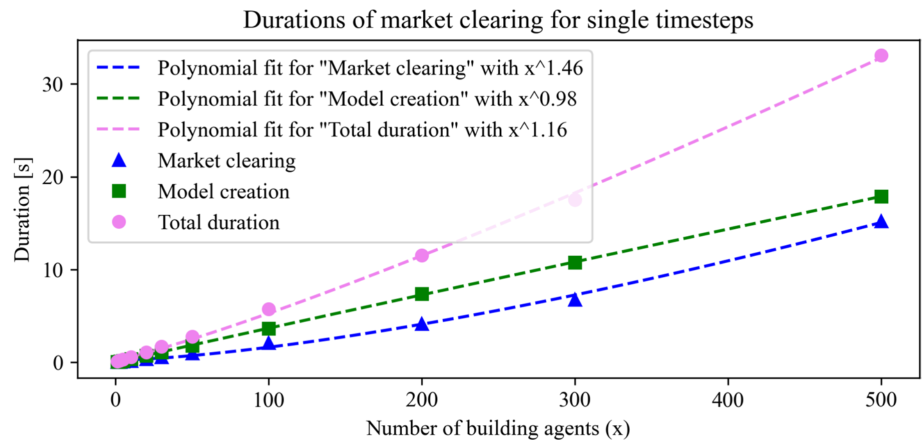

Fig. 4 The total durations of the market clearing (creation and solving of MILP-Problem) for different numbers of building-agents, averaged over four consecutive starting times. The lines indicate polynomial fits of durations for each number of agents. "Market clearing" is the wall clock time including transferring the problem and the solution between the solver and PYOMO. "Model creation" is the wall clock time as needed in PYOMO to formulate the problem. Additionally, the sum of the two is also shown

package *highspy* in version 1.5.3 [32] on a system with 8 Intel CPU-cores from the Comet Lake generation. These numbers are based on averages over 4 consecutive starting time-steps.

As the total duration scales slightly higher than linear, duration might become a problem when using very high numbers of participating agents. For the use-case of a district energy-system, however, it is not expected to pose a problem.

If the total duration of the market mechanism (model creation and solving) would pose issues with real-time capability, the model creation could be improved. The structure of the model rarely changes and therefore, the model often could be reused with only updating the parameters. Currently, the model is created from scratch every time-step to make sure it is up-to-date and correctly reflects the replies of the agents. Besides the number of agents, the number of options replied from each agent can differ each time the market clearing is performed. The forecast of each agent (and therefore the bids handed in) always span 96 time-steps of 15 minutes each, leading to a 24-hour horizon.

Additional modes of operation

As already stated in chapter 4, MASSIVE can be used to compare different modes of operation. This will be done in the upcoming chapter.

In the examinations and descriptions above, a single shot approach is used. There, the market is cleared only once for each starting time-step. Another approach is to clear the market multiple times per starting time-step, therefore following an iterative approach. By this, the agents can get an updated price multiple times to adapt their behavior on the new information. By redoing this many times iteratively a new solution is found that brings all agents near to a new equilibrium. To decide on whether the new solution settled near an equilibrium, the latest four solution are checked. This check is twofold. The relative deviation between two consecutive solutions must not be greater than a threshold (here, 1 ‰) and the deviation of all solutions from the average must also not be greater than the threshold (also 1 ‰). The latter constraint aims to avoid false positives in cases of drift. The use of relative thresholds is important, as the metrics (e.g. social welfare or traded power) are strongly dependent on the number of agents. Therefore, absolute thresholds do not work well when varying the number of participating agents. The settling near an equilibrium could be regarded as convergence, although it is no convergence that can be proven mathematically.

This reached equilibrium can be seen as *Nash equilibrium* in the context of game theory (see [33, 34] for more information on Nash equilibria).

The iterative approach can be pursued in two ways. Either all agents can update their forecast in parallel, or they update their forecast one after the other (in serial). Of course, also mixed forms are possible where agents are grouped and all members of the same group update in parallel, while the groups update after each other. A visualization of the two approaches (iterative parallel and iterative serial) is shown in Figure 8 in Appendix A. The change of modes is rather easy, as the agents simply respond to a request for a forecast. Therefore, the central communication can either request a single agent, multiple agents or all agents to provide an updated forecast. By this, the serial iterative, mixed or parallel iterative approach can be run without changing anything within the agents, as the whole coordination is done external of the agents. Because of this design decision, the agents are also easier to set up as they only react to requests, but do not need to become active by themselves.

If the results from market clearing get stable enough, the decision is communicated to all agents to put the offered actions into practice. Another reason for the iterative approach to end is if the iterations took too much time (because the market clearing was slow or the agents took much time to respond). Then, the approach needs to exit early, so that a (suboptimal) result can be communicated to the participants. This is required to allow real world applications, where a decision needs to be known in time for the next time-step to be incorporated. However, in all run tests, the iterations always ended earlier than required for real-time applications.

The technical part of the communication is the same for all approaches (iterative and single shot). The only difference is the number of market clearings and requests for forecasts from the agents, that are done for each individual time-step.

In the next parts, the differences between the single shot approach and the iterative approaches are explained and studied.

Determination of power prices

In comparison to the single shot approach, the iterative approach has a uniform pricing only for the final price (when the agents get the info on which option to implement). During all rounds of iteration to find the equilibrium, the prices are different for each agent. In the examples shown below, often five rounds were needed.

This difference comes from a penalty-term that is calculated per agent and per round of iteration and added to the resulting price of the market clearing. Adding a penalty term to make desired results more attractive is often used in optimization problems. The reason for this penalty term is to nudge the agents towards the known solution, that has the best overall social welfare. The market clearing is done identically to the market clearing in the single shot approach.

The equations for the new power price per agent is given in Equations (11) to (13) below.

There, the iteration is denoted as $k \in K$ starting at 0 with every new time-step. The agents are denoted as $a \in A$ and the time-steps are denoted as $t \in T$ as in the equations above.

$PR_{a,t}^{k+1}$ is the price for agent *a* at time-step *t* in iteration $k+1$. $PR_{MC,t}^{k}$ denotes the price derived by the market clearing during iteration *k* for time-step *t*. $PR_{\text{penalty},a,t}^{k}$ is an agent-specific penalty-term that differs for each iteration *k* of time-step *t*.

$lambda_k$ is a penalty-factor with α and β as parameters, whose influence will be examined in the following subsection.

$$PR_{a,t}^{k+1} = PR_{MC,t}^{k} + PR_{\text{penalty},a,t}^{k} \qquad \forall k \in K, \forall a \in A, \forall t \in T \tag{11}$$

$$PR_{\text{penalty},a,t}^{k} = \lambda_k \cdot \Delta rev_{a,k,t} \qquad \forall k \in K, \forall a \in A, \forall t \in T \tag{12}$$

$$\lambda_k = \max\left(\lambda_{k-1} \cdot \alpha + \beta\,,\, 0\right) \qquad \forall k \in K \tag{13}$$

The penalty term consists of two parts (cf. Equation 12). One part is the factor λ_k, that only depends on the current iteration *k*. The other part is the difference of revenue ($\Delta rev_{a,k,t}$) for that agent *a* at iteration *k* and time-step *t* compared the revenue of the same agent and time-step in the solution with the best social welfare. So ($\Delta rev_{a,k,t}$) has a different value for each time-step of the forecast, each agent and each iteration.

The factor λ_k decreases with increasing number of iterations (meaning $\lambda_k < \lambda_{k-1}$ in Equation (13)), but is the same for all time-steps *t* of a forecast and all agents *a*.

The idea of using a decreasing penalty is to nudge the agents to the best known solution at the beginning, but let them adapt to the updated results of the other agents towards the end. This is important as the final result (after the iteration converged) is calculated without any penalty term to achieve uniform pricing. For decreasing functions (towards zero), a linear decrease and an exponential decrease are commonly used ones. These two approaches are represented in the parameters α and β.

The choice of parameters α and β might is relevant for the development of the penalty-term during multiple iterations, and will be discussed in the next section.

As said above, the penalty term is only added to find the equilibrium. Once, this is reached, it is discarded and the power price is the same for all agents when they get the final result for the upcoming time-steps. This means there is uniform pricing for the final result, but not for the intermediate ones.

Influence of parameters on convergence

As said above, the choice of parameters $\lambda_{\text{init}}, \alpha$, and β is relevant for how the penalty develops over multiple iterations and therefore might influence the solution towards which the system converges.

In order decreasing λ_k with every iteration, the linear or exponential approach can be used (or combinations of those). Therefore, the following values satisfying the following restrictions for alpha and beta are examined: $0 < \alpha \leq 1$ and $\beta \leq 0$.

Figure 5 shows the influence of λ_{init} and α on the result after convergence compared to the result obtained by the single shot approach. Examinations not visualized here

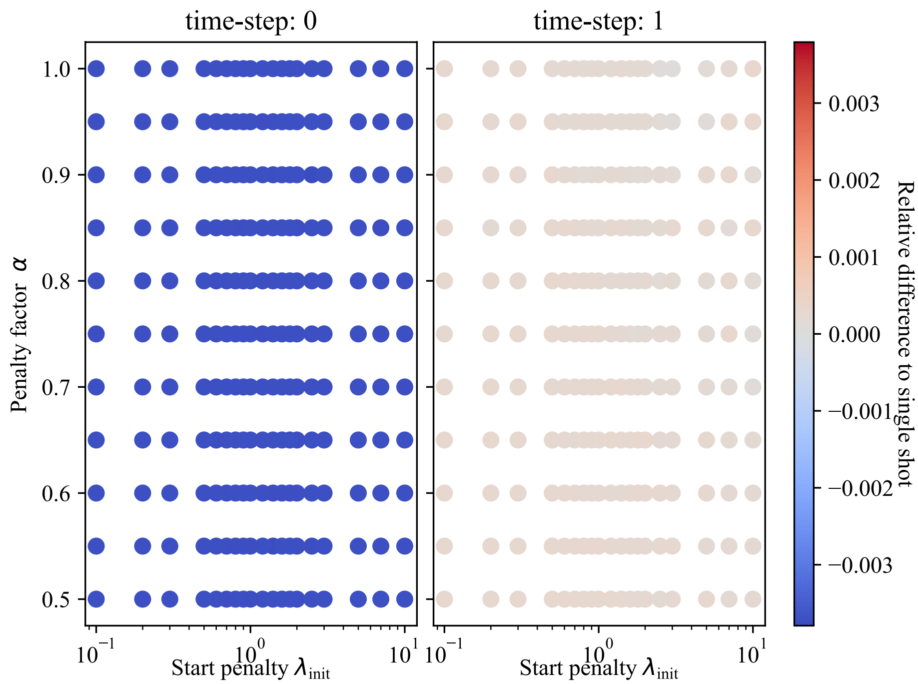

Fig. 5 Convergence results for varying penalty parameters when following a parallel iterative approach. For the initial time-step (left), it can be seen, that the iterative approach performs slightly better, than the single shot approach (indicated by the blue color). For the later time-steps (exemplarily shown on the right) this inverts and the iterative approach performs worse

show, that β should be set to zero, as all lower values make the result worse. Therefore, all examinations shown here use $\beta = 0$, meaning no absolute penalty, but only penalty factor.

In Figure 5 it can be seen, that the results are fairly similar. The relative differences are below 1 ‰ and therefore negligible for most real-world applications. There, values below zero (depicted in blue) represent a better objective than it is achieved in the single shot approach. Values above zero (shown in red) mean a worse result than in the single shot approach. However, for the first time-step, the iterative approach is slightly better than the single shot one. In the later time-steps this changes and the single shot approach outperforms the iterative one. This can be interpreted, that the iterative approach is more helpful in situations, where large differences between initial prices and traded volumes and final prices and volumes are to be expected. The result is similar if the examination is repeated with an energy system also including battery storage agents (not shown in this publication). There, the market clearing in the first time-step also benefits from the iterative approach, while the results for the later time-steps are in favor of the single-shot approach.

With regard to computational costs it can also be concluded that the iterative approach is not cheaper than the single shot one, as the determination of convergence needs at least four rounds of iterations. As a single round of iteration is computationally comparable to a single option in the single shot approach, it can be said, that the two approaches are at best similarly intensive in terms of computation. If more than the minimal four rounds of iteration are needed, the iterative approach needs more computational power than the single-shot approach with four options.

Therefore, it can be concluded, that the two approaches return comparable results with a slight advantage to the single-shot approach for most of the time-step. With regard to computational resources, the single-shot approach is also favorable.

As can be seen in Figure 5 by the nearly identical color for all parameter combinations, the choice of values for the parameters α and λ_{init} does not influence the result in any relevant way.

Iterative mode: serial, parallel, and hybrid

In the iterative, parallel approach, all agents are updated before a new market clearing takes place. As the market is only cleared again, after all agents are updated, the updates of the agents do not influence each other during the updating process. Therefore, the updating can be done in parallel. Another mode has been implemented in MASSIVE, where the market is cleared after a single agent updated its forecast. This approach is called iterative serial approach in the following. Then, the next agent is queried with the new results after the market clearing. Therefore, the agents cannot determine their forecast in parallel. Lastly, a hybrid mode is possible, where the agents are split into multiple groups, where all participants of a group are queried in parallel, but the multiple groups are queried sequentially. A group size of one is identical to the serial approach, whereas if all agents are part of the same group it is the same as the parallel approach. This helps to bridge the binary view of fully serial or fully parallel. The reason for these additional modes is that often, serial approaches are used (see e.g. [17, 35]). Direct comparisons of serial and parallel iterative approaches for the same energy system under investigation are not known to the authors. [10] compares those two approaches, but this is done for transportation problems and without the usage of market-mechanisms. For energy systems, to the best of the authors' knowledge, this work offers the new possibility to compare these approaches for distributed control with many agents coordinating themselves by the means of market-mechanisms.

The influence of the number of agents per group is small, as can be seen in Figure 6. There, the following parameters for convergence are chosen: $\alpha = 0.95$, $\beta = 0$, and $\lambda_{\text{init}} = 1$

There it is shown, that the smaller groups perform slightly worse than the larger groups (lower objectives mean better results). The improvement, however, is so small, that it is not relevant for most use cases (around 0.7 ‰). The single-shot approach also outperforms the parallel iterative approach. At least this holds true for the later time-steps, where differences between initial and final result of each round are small.

With regard to the computational effort it can be said, that with smaller group size (fewer agents in parallel), the market needs to be cleared more often. As no agent can update while the market is cleared, the whole process gets less parallel with reduced group size. This leads to a higher overall duration for each iteration and therefore for the overall convergence towards a solution.

An analysis of the influence of parameters on the convergence was also done for the iterative serial approach (as compared to the iterative parallel approach shown in the previous part). The meaning of the parameters α and λ_{init} is the same in the following Figure 7 as for Figure 5 above and also $\beta = 0$ is true here.

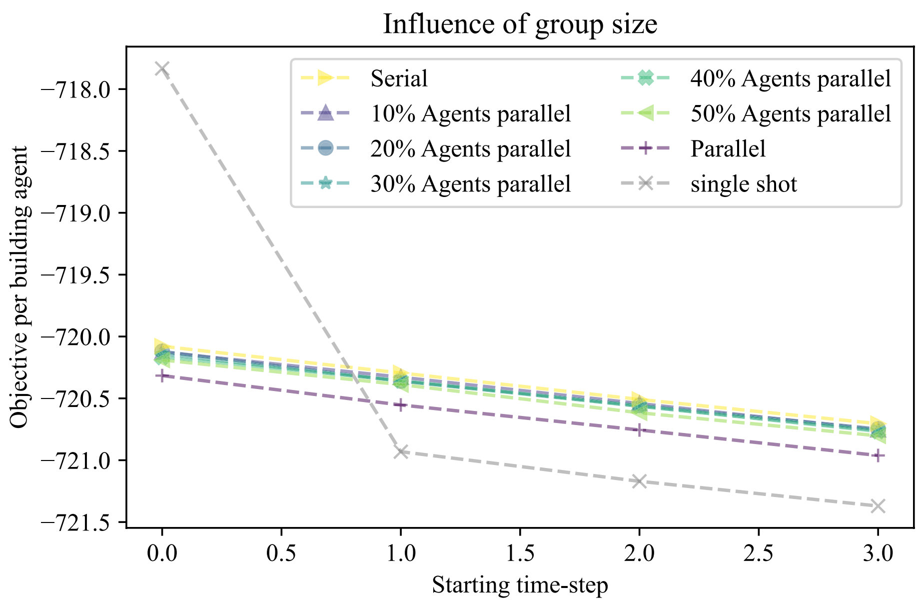

Fig. 6 Influence of the number of agents per group on the overall optimality. Lower values indicate a better result

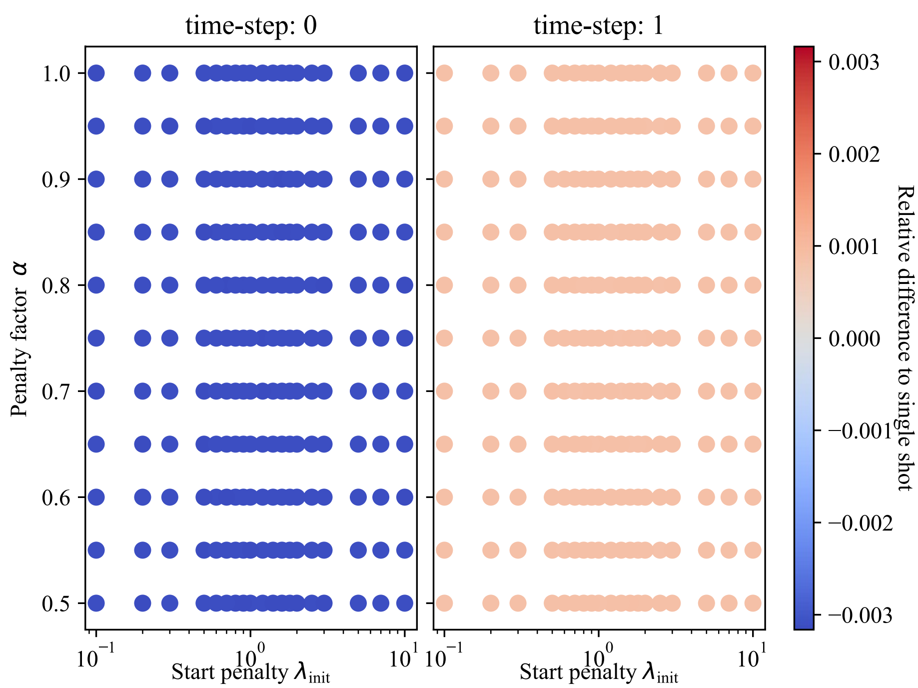

Fig. 7 Convergence results for varying penalty parameters when following a serial iterative approach. For the initial time-step (left), it can be seen, that the iterative approach performs slightly better, than the single shot approach (indicated by the blue color). For the later time-steps (exemplarily shown on the right) there is nearly no difference (depicted by the gray color)

Conclusion and outlook

This publication introduced a new agent-based approach for distributed control of a micro-grid. This control is aimed at time-scales in the scheduling region (e.g. time-steps of some minutes with horizons of a few hours). Examples are shown for time-steps of 15 minutes with a horizon of 24h. It uses a market clearing mechanism to coordinate supply and demand of power. The approach is modular and allows for easy addition or removal of agents. Since agents are decoupled from each other it is also possible design them individually and to update them as required.

The market scales well and is cleared below one minute even for many agents (examined up to 1500 agents) leading to the possibility to use this approach not only for small-sized systems.

Upcoming topics for further examinations include the usage of long-term storages (e.g. hydrogen) in the system, as well as including fees in the power options of the agents. These fees could be fixed if the option is chosen and not depend on the power traded (similar to a basic fee). Another advancement could be the addition of a grid topology in the market-mechanism.

With regard to the iterative modes it can be stated that it can perform better than the single-shot approach in situations/time-steps, where large changes are to be expected (e.g. the first step after initialization). In other cases, the single-shot approach performs similarly well or even slightly better. Furthermore, the single-shot approach often needs less computational resources, as the market clearing is done only once per time-step.

In a further differentiation, it can be said, that a parallel iterative approach often performs slightly worse than a serial iterative approach. But these differences are often negligible. Nevertheless, the parallel iterative approach is computationally less expensive and therefore preferable compared to the serial iterative approach.

In general, it can be said, that the iterative approaches are useful if the computational resources and durations are not a major constraint and if large deviations in the power supply and demand between consecutive time-steps (i.e. a very dynamical system) are to be expected. In most other cases, the single-shot approach should be used, as it requires less resources and performs similarly well.

With the shown examinations, this publication aims to help make design choices for own approaches of managing district energy-systems.

Appendix A

Visualization of iterative approaches

Appendix figure Fig. 8

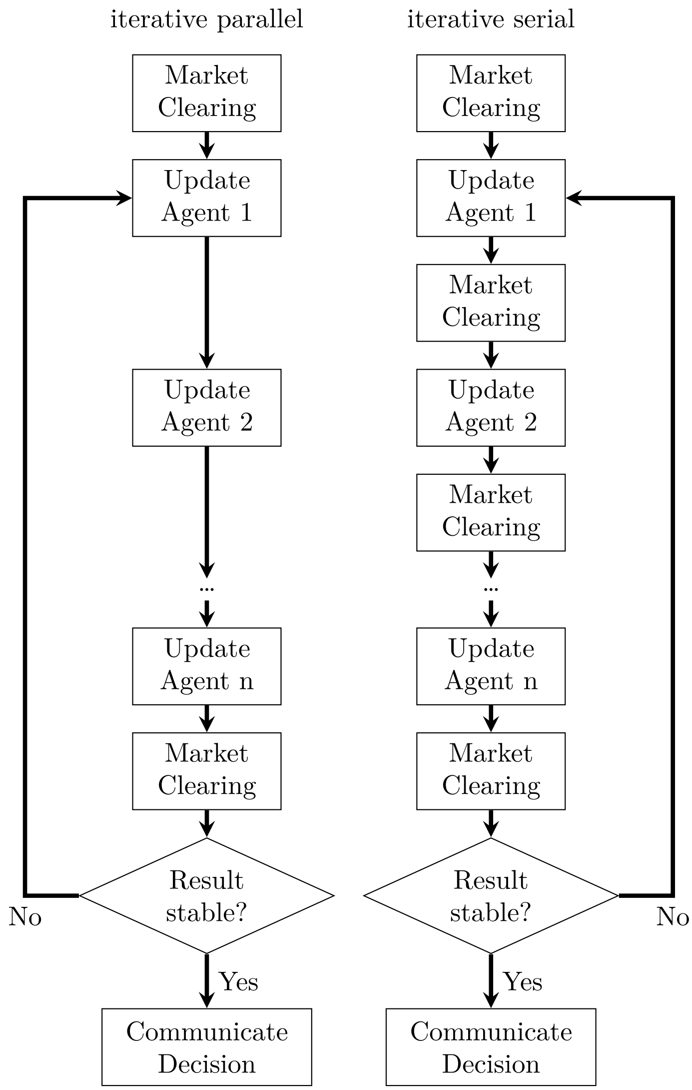

Fig. 8 The two iterative approaches introduced in this publication differ in when a market clearing takes place. In the iterative parallel approach, all agents are queried and update before the market clearing takes place. In the iterative serial approach, the market is cleared after a single agent updated its forecast. After all agents updated, it is checked if the result is similar enough to the ones from previous iterations. If so, the decision from the market clearing is communicated. If not, another round of iterations is run

Abbreviations

DSM	Demand Side Management
EPEX	European Power Exchange
MASSIVE	Multi Agent Scheduling Solution In a Virtual Environment
MILP	Mixed Integer Linear Problem
MINLP	Mixed Integer Nonlinear Problem
MPC	Model Predictive Control
PV	Photovoltaic
TMO	Test Meteorological Years

Acknowledgements
The present contribution is supported by the Helmholtz Association under the Joint Initiative "Energy Systems Integration".

Author contributions
JMF developed the concept and implemented the developed framework and analysis. He was the major author to the manuscript. LR assisted in the concept and implementation of the developed framework. AX worked on the concept of the developed framework and reviewed the manuscript. DM acquired the funding and reviewed the manuscript. All authors read and approved the final manuscript.

Funding
Open Access funding enabled and organized by Projekt DEAL. The development of the framework was supported by the Helmholtz Association, grant number DBA01529.

Data availability
The data can be generated with the provided code (see below) and are available from the corresponding author on reasonable request.

Code availability
The code used for the creation and analysis of the data underlying this manuscript is available publicly in the following repository: https://jugit.fz-juelich.de/iek-10/public/optimization/massive/-/tree/massive_paper In addition, the code is available as artifact of DOI 10.5281/zenodo.15768504.

Declarations

Competing interests
The authors declare that they have no competing interests

Received: 7 April 2025 / Accepted: 18 July 2025
Published online: 28 July 2025

References

1. Baldea M, Harjunkoski I (2014) Integrated production scheduling and process control: a systematic review. Comput Chem Eng 71:377–390. https://doi.org/10.1016/j.compchemeng.2014.09.002
2. Daoutidis P, Lee JH, Harjunkoski I, Skogestad S, Baldea M, Georgakis C (2018) Integrating operations and control: a perspective and roadmap for future research. Comput Chem Eng 115:179–184. https://doi.org/10.1016/j.compchemeng.2018.04.011
3. Seborg E (2011) Mellichamp, doyle: process dynamics and control. John Wiley and Sons Inc, Hoboken, New Jersey, USA
4. Scattolini R (2009) Architectures for distributed and hierarchical model predictive control - a review. J Process Control 19(5):723–731. https://doi.org/10.1016/j.jprocont.2009.02.003
5. Kumar J, Agarwal A, Agarwal V (2019) A review on overall control of DC microgrids. J Energy Storage 21:113–138. https://doi.org/10.1016/j.est.2018.11.013
6. Keshta HE, Ali AA, Saied EM, Bendary FM (2019) Real-time operation of multi-micro-grids using a multi-agent system. Energy 174:576–590. https://doi.org/10.1016/j.energy.2019.02.145
7. McArthur SDJ, Davidson EM, Catterson VM, Dimeas AL, Hatziargyriou ND, Ponci F, Funabashi T (2007) Multi-agent systems for power engineering applications-part I: concepts, approaches, and technical challenges. IEEE Trans Power Syst 22(4):1743–1752. https://doi.org/10.1109/tpwrs.2007.908471
8. Dimeas A, Hatziargyriou N (2004) A multi-agent system for microgrids. In: Methods and Applications of Artificial Intelligence, Springer, Berlin, Heidelberg, pp 447–455 https://doi.org/10.1007/978-3-540-24674-9_47
9. Christofides PD, Scattolini R, Peña DM, Liu J (2013) Distributed model predictive control: a tutorial review and future research directions. Comput Chem Eng 51:21–41. https://doi.org/10.1016/j.compchemeng.2012.05.011
10. Negenborn RR, De Schutter B, Hellendoorn H (2006) Multi-agent model predictive control of transportation networks. In: 2006 IEEE International Conference on Networking, Sensing and Control. IEEE, Ft. Lauderdale, FL, USA https://doi.org/10.1109/icnsc.2006.1673161
11. Negenborn RR, Schutter BD, Hellendoorn H (2006) Multi-agent model predictive control for transportation networks: serial versus parallel schemes. IFAC Proc Vol 39(3):339–344. https://doi.org/10.3182/20060517-3-fr-2903.00183
12. Liu C, Li Z (2021) Comparison of centralized and peer-to-peer decentralized market designs for community markets. IEEE Trans Ind Appl 58(1):67–77. https://doi.org/10.1109/TIA.2021.3119559
13. Ygge F, Akkermans H (1999) Decentralized markets versus central control: a comparative study. J Artif Intell Res 11:301–333. https://doi.org/10.1613/jair.627
14. Liu P, Ozguner U (2018) Distributed model predictive control of spatially interconnected systems using switched cost functions. IEEE Trans Autom Control 63(7):2161–2167. https://doi.org/10.1109/TAC.2017.2757881
15. Liu Q, Abbas HS, Mohammadpour J, Wollnack S, Werner H (2016) Distributed model predictive control of constrained spatially-invariant interconnected systems and in input-output and form. In: Proceedings of the American Control Conference (ACC), Boston Marriot Copley Place, Boston, MA, USA
16. Stadler P, Ashouri A, Maréchal F (2016) Model-based optimization of distributed and renewable energy systems in buildings. Energy Build 120:103–113. https://doi.org/10.1016/j.enbuild.2016.03.051
17. Kolen S, Molitor C, Wagner L, Monti A (2017) Two-level agent-based scheduling for a cluster of heating systems. Sustain Cities Soc 30:273–281. https://doi.org/10.1016/j.scs.2017.01.014
18. Riebesel LJ (2018) Evaluation of cloud-based energy flexibility services using an intelligent energy management system in a local district. Master's thesis, RWTH Aachen University - Faculty of Electrical Engineering and Information Technology - Institute for Automation of Complex Power Systems
19. Xu S, Zhao Y, Li Y, Zhou Y (2021) An iterative uniform-price auction mechanism for peer-to-peer energy trading in a community microgrid. Appl Energy 298:117088. https://doi.org/10.1016/j.apenergy.2021.117088
20. Boudoudouh S, Maâroufi M (2018) Multi agent system solution to microgrid implementation. Sustain Cities Soc 39:252–261. https://doi.org/10.1016/j.scs.2018.02.020
21. Yun H, Shim H, Ahn H-S (2019) Initialization-free privacy-guaranteed distributed algorithm for economic dispatch problem. Automatica 102:86–93. https://doi.org/10.1016/j.automatica.2018.12.033
22. EPEX Spot SE (April 2025) EPEX Spot Operational Rules. Techreport

23. Langiu M, Shu DY, Baader FJ, Hering D, Bau U, Xhonneux A, Müller D, Bardow A, Mitsos A, Dahmen M (2021) COMANDO: a next-generation open-source framework for energy systems optimization. Comput Chem Eng 152:107366. https://doi.org/10.1016/j.compchemeng.2021.107366
24. Hart WE, Watson J-P, Woodruff DL (2011) Pyomo: modeling and solving mathematical programs in Python. Math Program Comput 3(3):219–260. https://doi.org/10.1007/s12532-011-0026-8
25. Hart WE, Laird CD, Watson J-P, Woodruff DL, Hackebeil GA, Nicholson BL, Siirola JD (2017) Pyomo– optimization modeling in python, vol 6330. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-319-58821-6
26. Modelon: JModelica. https://jmodelica.org/ Accessed 2020-03-03
27. Gómez EA, Conejo AJ, Cañizares C (2018) Electric energy systems: analysis and operation, 2nd edn. CRC Press, Taylor & Francis Group, Boca Raton
28. Mas-Colell A, Whinston MD, Green JG (1995) Microeconomic theory. Oxford University Press, New York, Oxford
29. Baader FJ, Mork M, Xhonneux A, Müller D, Bardow A, Dahmen M (2020) Mixed-integer dynamic scheduling optimization for demand side management. In: Proceedings Of The 30th European Symposium On Computer Aided Process Engineering (escape30) https://doi.org/10.1016/B978-0-12-823377-1.50235-4
30. Holmgren WF, Hansen CW, Mikofski MA (2018) Pvlib python: a python package for modeling solar energy systems. J of Open Source Softw 3(29):884. https://doi.org/10.21105/joss.00884
31. Holmgren W, Calama-Consulting Hansen C, Anderson K, Mikofski M, Lorenzo A, Krien U, Bmu SC, DaCoEx DA, Jensen AR, De León Peque MS, Konstant_t, Mayudong, Heliolytics ME, Anoma MA, Guo V, Boeman L, Stein J, Vining W, Jforbess LT, Morgan A, Ranalli J, Leroy CRAM, JPalakapillyKWH, Dollinger J (2021) Pvlib/Pvlib-Python: V0.9.0. Zenodo. https://doi.org/10.5281/ZENODO.5366883. https://zenodo.org/record/5366883 Accessed 2021-11-10
32. Huangfu Q, Hall JAJ (2018) Parallelizing the dual revised simplex method. Math Program Comput 10(1):119–142. https://doi.org/10.1007/s12532-017-0130-5
33. Nash JF (1950) Equilibrium points in n-person games. Proc Natl Acad Sci 36(1):48–49. https://doi.org/10.1073/pnas.36.1.48
34. Nash J (1951) Non-cooperative games. The Ann of Math 54(2):286. https://doi.org/10.2307/1969529
35. Trodden P, Richards A (2006) Robust distributed model predictive control using tubes. In: 2006 American Control Conference, p. 6. IEEE, Minneapolis, MN, USA. https://doi.org/10.1109/ACC.2006.1656519. http://ieeexplore.ieee.org/document/1656519/ Accessed 2022-04-20

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.