Electrical Power and Energy Systems 133 (2021) 107228

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

A mixed-integer distributed approach to prosumers aggregation for providing balancing services

Alessio La Bella [*], Alessandro Falsone, Daniele Ioli, Maria Prandini, Riccardo Scattolini

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, via Ponzio 34/5, 20133 Milano, Italy

ARTICLE INFO

Keywords:
Balancing services
Prosumers aggregation
Distributed MILP optimization

ABSTRACT

This paper addresses the provision of ancillary services in smart energy systems. A large number of prosumers are aggregated by an Energy Service Provider (ESP) in order to provide a manual Frequency Restoration Reserve (mFRR) service, which consists in offering some degree of flexibility and be willing to provide a power variation over a given time interval upon reception of an explicit manual request by the Transmission System Operator (TSO). The main focus of this paper is to define how the ESP can optimally distribute the requested flexibility effort to the prosumers in the pool, promptly providing the agreed mFRR service upon request of the TSO. In particular, a scalable strategy is proposed, able to account for integer decision variables like on/off commands, while reducing the combinatorial complexity of the problem and preserving privacy of local information via distributed computations. Lead and rebound effects are avoided by maintaining the originally scheduled energy exchange profile before and after the time interval where the TSO request must be satisfied. The simulation results show the effectiveness of the proposed approach in terms of scalability and quality of the obtained feasible solution.

1. Introduction

The wider and wider penetration of Renewable Energy Sources (RES) in the electricity world is motivated by many advantages they can guarantee in economic terms, for pollutants reduction, and for efficient management of the electricity grid. For these reasons, RES, together with the proper management of programmable loads, are today the keystone for the creation of an electricity network no longer based on a few large plants, but made by many distributed energy generators, energy storage systems, and controllable loads. This new paradigm provides high flexibility at the price of new coordination problems, due to the presence of non-programmable elements like non dispatchable energy sources and loads, and to the need to guarantee reliable ancillary services, like power production and frequency/voltage regulation.

To cope with these new challenges, distributed energy resources can be used as external balancing services, [1]. Indeed, an active participation of the so-called *prosumers*, like microgrids and smart buildings, is nowadays recognized as one of the best solutions to provide additional flexibility to electricity markets, [2]. However, due to their (often) small-scale size, in order to reach an adequate power capability level, multiple prosumers must be coordinated by an *aggregator*, named here Energy Service Provider (ESP) [3,4]. Aggregators have been widely studied in recent years, see [5,6], and methods for assessing and optimizing the flexibility that can be obtained through aggregation of multiple energy systems for the day-ahead market operations are presented in [7–10].

The ESP can also contribute to the manual Frequency Restoration Reserve (mFRR) service (traditionally named *Tertiary Reserve*) and provide active power services for restoring frequency deviations, [11,6]. To participate to this service, the ESP must communicate its power availability in advance, submitting upward and downward reserve bids to proper reserve markets. Reserve markets have still not reached the harmonization and standardization level of other energy markets in Europe (e.g. the day-ahead market) and there are still differences among European countries [12]. According to MIBEL market, which covers Spain and Portugal control areas, mFRR bids are submitted by providers the day prior to real-time operation, until 21:00 [13]. In some other countries, mFRR bids can be submitted also during the daily operation, e.g. in Norway up to 45 min before the effective delivery [14], while in The Netherlands up to 15 min [11]. In Italy, the reserve market is split in

[*] Corresponding author.
E-mail addresses: alessio.labella@polimi.it (A. La Bella), alessandro.falsone@polimi.it (A. Falsone), daniele.ioli@polimi.it (D. Ioli), maria.prandini@polimi.it (M. Prandini), riccardo.scattolini@polimi.it (R. Scattolini).

https://doi.org/10.1016/j.ijepes.2021.107228
Received 6 May 2020; Received in revised form 31 March 2021; Accepted 20 May 2021
Available online 15 June 2021

two: an ex-ante reserve market, where mFRR bids can be submitted until 17:30 of the day prior to real-time operation, and an online reserve market, where providers can update their offers every 4 h during the daily operation [15].

In general, after the gate closure time for bids submission, the Transmission System Operator (TSO) defines which offers to accept based on a precise merit order list [16]. The providers are typically remunerated for the offered mFRR services, but different market frameworks apply also for this aspect. For instance, in some countries the delivered upward and downward mFRR services are remunerated with the pay-as-bid rule (e.g in Germany, Finland and Italy) while in others they are remunerated with common marginal prices (e.g. in Norway, Netherlands, Spain and Portugal) [13,14,11,15]. For more many additional details on mFRR market frameworks, the interested reader is referred to [11,17,14,6].

Independently on the local market regulations, the TSO can then request the ESP to effectively deliver the offered mFFR services during the online operation, asking to vary the active power profile of its prosumers pool with respect to the pre-agreed power baseline for a specific time period [18]. The TSO requests can be also a fraction of the offered mFRR bids, based on the actual needs of the electrical system [19]. As dictated by the European regulations [16], the ESP must deliver the requested mFRR service within 15 min, optimally rescheduling the operations of the prosumers pool independently on its size.

In view of the above considerations, there is therefore the need to devise efficient strategies for the ESP to promptly respond to TSO power variation requests, while also maintaining the pre-agreed power baseline before and after the satisfaction of the request, thus avoiding the so-called *lead* and *rebound effects*, see [2,20]. The design of proper optimization algorithms is however made difficult by the fact that prosumers, i.e. dispatchable generators, storage systems, and variable loads, can have a discontinuous behavior, for instance generators can be switched on/off, and controllable loads can be shifted, interrupted or curtailed. When adopting optimization approaches, this leads to the introduction of Mixed-Integer (MI) decision variables, so that the resulting optimization problem turns out to be a non convex MI program.

The solution of aggregators' optimization problems can be addressed with *centralized* algorithms, as described in [21,4,22], where the problem of jointly coordinating a group of smart houses, buildings or microgrids is considered. Centralized methods can in principle provide optimality guarantees, but can easily lead to scalability, computational, and confidentiality issues. To overcome computational limitations, centralized evolutionary algorithms for MI problems are proposed in [23,24], sacrificing optimality and without guaranteeing privacy of local information. For these reasons, recent research efforts have been devoted to the design of *distributed* optimization methods, enabling prosumers to locally optimize the operation of their own units, while cooperating to determine the optimal global solution. Standard distributed optimization methods may lead to feasibility and optimality issues when applied to MI (non-convex) problems, and therefore novel techniques have been recently proposed (see [25] for an in-depth analysis of the literature). In particular, a distributed approach is presented in [26], where individual users solve local MI problems through heuristic greedy algorithms leading to approximate solutions. Distributed Lagrangian-based algorithms have been developed and applied to the field, such as in [27,28], where however feasibility is not guaranteed and it is achieved by resorting to heuristic approaches. Distributed algorithms relying on the Dantzig-Wolfe (DW) decomposition method have been used for coordinating aggregators in [29,30], and applied to specific classes of systems and not to MI problems. In [25], a more general algorithm for MI linear problems has been developed, still relying on the DW method, without however considering the ancillary services provision problem. In this context, an original solution has recently been proposed in [31] based on convex relaxation and a distributed stochastic dual gradient algorithm. However, the mentioned approaches do not address the problem of the real-time mobilization of upward or downward power reserve satisfying the TSO requests, which is the focus of this paper. Concerning this aspect, a two-stage optimization framework applied to communities of households is presented in [32], however considering just continuous variables. An approach based on a hierarchical control architecture for MI problems is described in [33], where at the high level, an optimizer defines the flexibility reserves, while at the low level, a simple controller tracks the power references mobilizing the required power reserve. The approaches for real-time reserve provision described in [33,32] are however based on centralized approaches, i.e. possibly leading to scalability and privacy issues.

Trying to fill in the existing lack of an appropriate methodology to address optimization problems arising in prosumers' aggregation for mFRR provision, this work proposes a novel and effective distributed optimization algorithm for the solution of MI Liner Programming (MILP) problems which enables the prompt mobilization of power reserve in real-time by the ESP. The main contributions of our paper are:

- the MILP formulation of the ESP task to satisfy TSO power requests by properly coordinating prosumers, while avoiding lead and rebound effects;
- a privacy-preserving distributed coordination procedure that is scalable in the number of prosumers and mitigates the combinatorial complexity arising in mixed integer optimization.

It is assumed that the available flexibility reserve have been already traded by the ESP with the TSO in previous market stages, e.g. during the day-ahead as reported in [10,22,33]. The proposed algorithm is inspired by the distributed optimization method described in [34] for constraint-coupled Mixed-Integer Linear Programs (MILPs), which is based on the dual-decomposition approach and characterized by guaranteed finite-time convergence and feasibility properties, achieved through a tightening procedure of the coupling constraints. However, in some cases, the algorithm in [34] can be overly conservative, shrinking the feasibility region up to a point where no solution can be found. Therefore, to alleviate this issue, we propose here a variation of the algorithm in [34], which involves a more cautious tightening of the feasibility region based on the amount of violation of the coupling constraints. The proposed approach shows to be significantly efficient for the resolution of large-scale MILPs encountered in realistic applications. The extensive numerical tests show the effectiveness of the approach in determining a feasible solution close to the optimal one, allowing the ESP to promptly coordinate multiple prosumers to satisfy the TSO power request. Moreover, the experiments witness the enhanced scalability properties of the algorithm, as the computational time remains considerably smaller with respect to the 15 min time constraint imposed by the mFRR regulations, despite the possibly large size of the prosumers' aggregation.

The rest of the paper is structured as follows. In Section 2 the addressed problem is described and formulated as a MILP by suitably modeling the prosumers in the pool. The proposed distributed optimization strategy is described in Section 3. Numerical results are reported in Section 4 and some final conclusions are discussed in Section 5.

2. Problem description and MILP formulation

Consider an ESP that coordinates N prosumers providing balancing services to the grid. Each prosumer is equipped with different types of controllable devices: a programmable load Pl that can be modulated and shifted in time to a certain extent, a non-interruptible load Sl that can be only shifted, a controllable generator G (e.g., a microturbine), and a battery storage device B, which enhances its flexibility in terms of power exchange with the grid. In order to keep the notation light when formulating the coordination problem, in the following it is supposed that each prosumer has one device of each type. Indeed, the absence of a certain kind of device in a prosumer can be easily handled by omitting it, while the presence of multiple devices of the same type can be handled by indexing devices of the same type with a subscript. The optimization variables used in the following problem formulation are reported in Table 1.

It is assumed that the ESP has agreed with the TSO some reference daily power exchange profile and its availability to provide balancing services: if at some point during the day the grid is experiencing an imbalance between production and consumption, the TSO can ask the ESP to modify the reference power profile of the prosumers pool by a certain amount with some tolerance bounds, over a given time frame. Once the ESP has received such a request, the usage of the controllable devices of possibly all prosumers needs to be rescheduled so as to meet the request while, at the same time, minimizing the involved operating costs. It is assumed that the reference power profiles of the prosumers in the pool and the amount of flexibility offered to the TSO by the ESP are given (e.g., they are computed by suitable methods like that in [10]), as the designed strategy defines how the ESP can optimally respond to the TSO request through the solution of a Mixed-Integer Linear Program (MILP). The problem is formulated in centralized form to clearly describe the overall objective, decision variables, local and global constraints. By exploiting the partially decoupled structure of the global optimization problem, we propose in Section 3 a distributed computation solution where prosumers locally optimize their own units while the ESP coordinates their operations through an iterative negotiation mechanism (see Fig. 1). Conversely to centralized methods, this framework allows prosumers not to disclose their internal information to the ESP, and to promptly solve a large-scale MILP since prosumers perform their local operations in parallel.

Consider a one-day time frame discretized into M time slots of duration τ_s each. Reference is made in the following to the (average) power per time slot. Let t denote the index of the time slot from $t\tau_s$ to $(t+1)\tau_s, t = 0, \dots, M-1$. The power $P(t)$ produced ($P(t) > 0$) or absorbed ($P(t) < 0$) within time slot t by the pool of prosumers coordinated by the ESP is given by

$$P(t) = \sum_{i=1}^{N} P_i(t), \tag{1}$$

where $P_i(t)$ is the power generated/absorbed by prosumer i during time slot t with the same sign convention. For each prosumer i and each time slot t,

Table 1
Main optimization variables and parameters.

Symbol	Description	Unit
Γ	Power variation requested by the TSO to the prosumers pool	kW
P	Output power of the prosumers pool	kW
P_i	Output power of i-th prosumer	kW
P_i^G	Output power of the controllable generator of the i-th prosumer	kW
δ_i^G	On/Off status of the controllable generator of the i-th prosumer	Boolean
P_i^B	Output power of the battery of the i-th prosumer	kW
$P_i^{B,c}$	Charging power of the battery of the i-th prosumer	kW
$P_i^{B,d}$	Discharging power of the battery of the i-th prosumer	kW
δ_i^B	Charging/Discharging status of the battery of the i-th prosumer	Boolean
S_i^B	Energy stored in the battery of the i-th prosumer	kWh
P_i^{PL}	Power required by the programmable load of the i-th prosumer	kW
δ_i^{PL}	Power consumption level of the programmable load of the i-th prosumer	Integer
P_i^{SL}	Power required by the non-interruptible load of the i-th prosumer	kW
δ_i^{SL}	Activation of the non-interruptible load of the i-th prosumer	Boolean

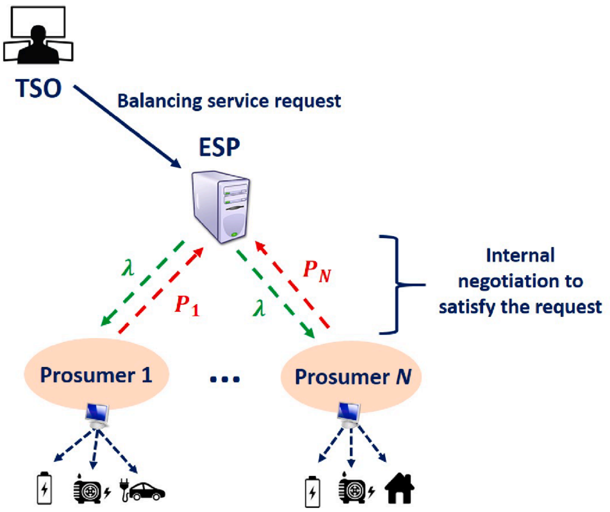

TSO
Balancing service request
ESP
λ
P_1
λ
P_N
Internal negotiation to satisfy the request
Prosumer 1
...
Prosumer N

Fig. 1. Pictorial view of the distributed implementation of Algorithm 1 by an ESP coordinating a pool of N prosumers for the satisfaction of a TSO request.

$$P_i(t) = P_i^G(t) + P_i^B(t) - P_i^{Pl}(t) - P_i^{Sl}(t), \tag{2}$$

where $P_i^G(t) \geqslant 0$ is the power produced by its controllable generator, $P_i^B(t)$ is the power exchanged when discharging ($P_i^B(t) > 0$) or charging ($P_i^B(t) < 0$) the battery, and $P_i^{Pl}(t) \geqslant 0$ and $P_i^{Sl}(t) \geqslant 0$ are the power requested by the programmable and the non-interruptible load, respectively. In case of multiple devices, the quantities on the right hand side of (1) shall be replaced by the summation over the respective device index.

The pre-agreed reference power profile of prosumers i is denoted as $\widetilde{P}_i(t)$ and can be decomposed in the contributions of the controllable devices G, B, Pl, and Sl : $\widetilde{P}_i(t) = \widetilde{P}_i^G(t) + \widetilde{P}_i^B(t) - \widetilde{P}_i^{Pl}(t) - \widetilde{P}_i^{Sl}(t)$. The reference power profile of the prosumers pool is then given by $\widetilde{P}(t) = \sum_{i=1}^{N} \widetilde{P}_i(t)$.

As detailed next, the decision variables involved in the power profile modulation at each single prosumer level per time slot are both discrete and continuous.

As for the controllable generator G, it can be either on or off. When it is on, the average power produced within any given time slot must be within a minimum $\underline{P}_i^G$ and a maximum $\overline{P}_i^G$ value.

The binary variable $\delta_i^G(t) \in \{0, 1\}$, denoting the status of G (1 being the on status and 0 the off status), is therefore linked to the power output $P_i^G(t)$ as follows

$$\begin{cases} P_i^G(t) \in \left[\underline{P}_i^G, \overline{P}_i^G\right] & \delta_i^G(t) = 1 \\ P_i^G(t) = 0 & \delta_i^G(t) = 0, \end{cases} \tag{3}$$

which can be conveniently rewritten as the following constraints

$$\delta_i^G(t)\underline{P}_i^G \leqslant P_i^G(t) \leqslant \delta_i^G(t)\overline{P}_i^G, \tag{4}$$

which is linear in $P_i^G(t)$ and $\delta_i^G(t)$.

Typically, when the generator is switched on, it must be kept running for at least $T_i^{G,u}$ consecutive time slots and, similarly, it must stay off for at least $T_i^{G,d}$ time slots when switched off. These minimum up/down times requirements on G are captured by the following implications

$$\begin{aligned} & \delta_i^G(t) = 0 \ \wedge \ \delta_i^G(t+1) \ = \ 1 \\ \rightarrow \ & \delta_i^G(t+1) = \cdots = \delta_i^G\left(t + \tau_i^{G,u}\right) = 0, \end{aligned} \tag{5a}$$

$$\begin{aligned} &\delta_i^G(t)=1 \wedge \delta_i^G(t+1)=0 \\ \rightarrow\ &\delta_i^G(t+1)=\cdots=\delta_i^G\left(t+\tau_i^{G,d}\right)=0, \end{aligned} \qquad (5b)$$

for $t=0,\ \ldots,\ M-2$, with $\tau_i^{G,u}=\min\{T_i^{G,u},M-1-t\}$ and $\tau_i^{G,d}=\min\{T_i^{G,d},M-1-t\}$ being equal to the desired time intervals $T_i^{G,u}$ and $T_i^{G,d}$, respectively, or the length of the residual time horizon. Logic conditions in (5b) can be easily imposed through the following inequalities

$$\tau_i^{G,u}\left(\delta_i^G(t+1)-\delta_i^G(t)\right)\leqslant\sum_{\tau=1}^{\tau_i^{G,u}}\delta_i^G(t+\tau) \qquad (6a)$$

$$\tau_i^{G,d}\left(\delta_i^G(t)-\delta_i^G(t+1)\right)\leqslant\sum_{\tau=1}^{\tau_i^{G,d}}\left(1-\delta_i^G(t+\tau)\right), \qquad (6b)$$

which are both linear in $\delta_i^G(t)$.

The battery storage device is modeled as an integrator, whose state $S_i(t)>0$ denotes the battery energy content at the beginning of time slot t ($S_i(M)$ being the battery energy content at the end of the day) and satisfies the following recursive equation

$$S_i(t+1)=S_i(t)-\tau_s\frac{1}{\zeta_i^d}P_i^{B,d}(t)-\tau_s\zeta_i^c P_i^{B,c}(t), \qquad (7)$$

where $P_i^{B,d}$ is the discharging power, $P_i^{B,c}$ is the charging power, and ζ_i^d, $\zeta_i^c\in(0,1)$ are respectively the discharging and charging efficiencies taking into account the corresponding losses. According to the adopted convention, the discharging power is defined as non-negative, i.e. $P_i^{B,d}(t)\geqslant 0$, while the charging power as non-positive, i.e. $P_i^{B,c}(t)\leqslant 0$. Therefore, by introducing a binary variable $\delta_i^B(t)\in\{0,1\}$ representing the operation mode of the battery (1 for discharging and 0 for charging), the following linear power exchange limits are enforced

$$\left(1-\delta_i^B(t)\right)\underline{P}_i^B\leqslant P_i^{B,c}(t)\leqslant 0 \qquad (8a)$$

$$0\leqslant P_i^{B,d}(t)\leqslant\delta_i^B(t)\overline{P}_i^B, \qquad (8b)$$

where $\overline{P}_i^B$ and $\underline{P}_i^B$ are the upper and the lower bound of the battery output power $P_i^B(t)$, respectively. The overall battery power output is then given by

$$P_i^B(t)=P_i^{B,d}(t)+P_i^{B,c}(t), \qquad (9)$$

where it is highlighted that $P_i^{B,d}(t)$ and $P_i^{B,c}(t)$ cannot be simultaneously different from zero given (8a).

Iterating (7), the value of $S_i(t+1)$ can be expressed as a function of $P_i^{B,d}(s)$ and $P_i^{B,c}(s), s=0,\ldots,t$ as follows

$$S_i(t+1)=S_i(0)-\tau_s\sum_{s=0}^{t}\left(\frac{1}{\zeta_i^d}P_i^{B,d}(s)+\zeta_i^c P_i^{B,c}(s)\right), \qquad (10)$$

where $S_i(0)$ is the battery initial state. Clearly, the battery energy content $S_i(t+1)$ must be kept between a minimum $\underline{S}_i>0$ and a maximum $\overline{S}_i>\underline{S}_i$ energy level, which can be imposed by the following linear constraints on $P_i^{B,d}(s)$ and $P_i^{B,c}(s), s=0,\ldots,t$:

$$\underline{S}_i\leqslant S_i(0)-\tau_s\sum_{s=0}^{t}\left(\frac{1}{\zeta_i^d}P_i^{B,d}(s)+\zeta_i^c P_i^{B,c}(s)\right)\leqslant\overline{S}_i. \qquad (11)$$

As for the load, the programmable one *Pl* is firstly modeled, whose power consumption P_i^{Pl} can be modulated and/or deferred in time. Considering the realistic application, it is imposed that P_i^{Pl} cannot be continuously modulated but can operate only at specific levels, which correspond to certain fractions of its maximum power consumption $\overline{P}_i^{Pl}$ per time slot. Let $\delta_i^{Pl}(t)\in\{0,1,\ldots,n_i^{Pl}\}$ be a discrete variable denoting the level of consumption of *Pl* during time slot *t*. Then, the power consumption of *Pl* in time slot *t* can be expressed as

$$P_i^{Pl}(t)=\frac{\delta_i^{Pl}(t)}{n_i^{Pl}}\overline{P}_i^{Pl}, \qquad (12a)$$

$$\delta_i^{Pl}(t)\in\{0,1,\ldots,n_i^{Pl}\}. \qquad (12b)$$

The case where *Pl* cannot be modulated, is easily accounted for setting $n_i^{Pl}=1$. Typically, the flexibility offered by programmable loads is limited to a given time window. Moreover, it is important to ensure that *Pl* receives the overall amount of energy E_i^{Pl} required for its correct operation. These two conditions can be formulated in terms of the following constraints

$$P_i^{Pl}(t)=\widetilde{P}_i^{Pl}(t) \qquad t<t_i^{Pl,0}\ \vee\ t>t_i^{Pl,f}, \qquad (13a)$$

$$\sum_{t=0}^{M-1}\tau_s P_i^{Pl}(t)=E_i^{Pl}, \qquad (13b)$$

where $t_i^{Pl,0}$ and $t_i^{Pl,f}\geqslant t_i^{Pl,0}$ denote the first and the last time slots in which *Pl* can be modulated or shifted.

Finally, a non-interruptible/deferrable load *Sl* is considered, whose power profile cannot be modulated but only shifted in time. Real examples of this kind of loads are most industrial batch processes, which must absorb a precise power profile and they must be executed within a prescribed time window. Let τ_i^{Sl} denote the nominal time slot in which *Sl* is active and consider the binary variable $\delta_i^{Sl}(t)\in\{0,1\}$ which is equal to 1 if and only if the activation of *Sl* is shifted at time slot *t*. Moreover, let $t_i^{Sl,0}$ and $t_i^{Sl,f}$ be the first and final time slots in which *Sl* is allowed to be activated. Clearly, *Sl* has to be activated only once and within the time frame $[t_i^{Sl,0},t_i^{Sl,f}]$, and this condition can be modeled as

$$\delta_i^{Sl}(t)=0 \qquad t<t_i^{Sl,0}\ \vee\ t>t_i^{Sl,f}, \qquad (14a)$$

$$\sum_{t=t_i^{Sl,0}}^{t_i^{Sl,f}}\delta_i^{Sl}(t)=1. \qquad (14b)$$

Note that $t_i^{Sl,f}$ must be chosen accounting for the number of time slots needed by the load to complete its task. Once *Sl* has been activated at time slot *t*, it needs to follow its (shifted) scheduled power profile $\widetilde{P}_i^{Sl}(t)$, i.e.,

$$\delta_i^{Sl}(\tau)=1 \quad\Leftrightarrow\quad P_i^{Sl}(t)=\begin{cases}\widetilde{P}_i^{Sl}\left(t-\left(\tau-\tau_i^{Sl}\right)\right) & t-\left(\tau-\tau_i^{Sl}\right)\in[0,\ M-1]\\ 0 & otherwise\end{cases} \qquad (15)$$

for all $t=0,\ldots,M-1$, which is equivalent to the linear constraint

$$P_i^{Sl}(t)=\sum_{\tau=\tau_i^{Sl,\min}}^{\tau_i^{Sl,\max}}\widetilde{P}_i^{Sl}\left(t-\left(\tau-\tau_i^{Sl}\right)\right)\delta_i^{Sl}(\tau), \qquad (16)$$

where $\tau_i^{Sl,\min}=\max\{t+\tau_i^{Sl}-(M-1),0\}$ and $\tau_i^{Sl,\max}=\min\{t+\tau_i^{Sl},M-1\}$.

Suppose now that the ESP receives a request to modify the reference power exchange profile $\widetilde{P}(t)$ of the prosumers pool within the time frame from the time slot t_0 to the time slot t_f by an amount $\Gamma(t)\pm\varepsilon\Gamma(t), t\in[t_0,t_f]$, where $\varepsilon\in(0,1)$is a given relative tolerance parameter. It is assumed that the request is received at the beginning of time slot t_r satisfying $(t_0-t_r)\tau_s\geqslant 15$ minutes, and that the power variation window is synchronous with the adopted time discretization. Since the request has been received at t_r, the units' power profiles must not be modified for $t = 0, \dots, t_r$ and therefore it is imposed that

$$\begin{aligned} &P_i^G(t) = \widetilde{P}_i^G(t), \\ &P_i^B(t) = \widetilde{P}_i^B(t), \\ &P_i^{Pl}(t) = \widetilde{P}_i^{Pl}(t), \\ &P_i^{Sl}(t) = \widetilde{P}_i^{Sl}(t), \\ &t = 0, \dots, t_r. \end{aligned} \tag{17}$$

Then, the ESP has to reschedule each prosumer operations by suitably choosing the values of $P_i^G(t)$,$P_i^B(t)$,$P_i^{Pl}(t)$, and $P_i^{Sl}(t)$,$t = t_r + 1$,...,$M-1$, for all $i \in \mathscr{N} = \{1, \dots, N\}$, compatibly with all the previous constraints, so as to meet the TSO power variation request

$$\begin{aligned} &(1-\varepsilon)\Gamma(t) \leqslant P(t) - \widetilde{P}(t) \leqslant (1+\varepsilon)\Gamma(t), \\ &t = t_0, \dots, t_f, \end{aligned} \tag{18}$$

while satisfying the additional constraints

$$\begin{aligned} &P_i(t) = \widetilde{P}_i(t), \\ &t = t_r + 1, \dots, t_0 - 1, t_f + 1, \dots, M-1, \end{aligned} \tag{19}$$

for all $i \in \mathscr{N}$. Constraints (19) force each prosumer i to maintain its original power exchange profile before and after the satisfaction of the TSO request, so as to avoid the so-called *lead effect* and *rebound effect* outside the TSO power variation window.

Among the different solutions to satisfy the TSO request, the ESP chooses the optimal one by minimizing the following objective function

$$\begin{aligned} J(\cdot) = &\sum_{t=t_r+1}^{M-1} \sum_{i=1}^{N} \left(C_i^G P_i^G(t) + C_i^B \left| P_i^B(t) - P_i^B(t-1) \right| + C_i^{Pl} \left| P_i^{Pl}(t) - \widetilde{P}_i^{Pl}(t) \right| + C_i^{Sl} \left| \sum_{\tau=0}^{M-1} \tau \delta_i^{Sl}(\tau) - \tau_i^{Sl} \right| \right) \\ &- \sum_{t=t_r+1}^{M-1} p_R(t) \sum_{i=1}^{N} \left(P_i(t) - \widetilde{P}_i(t) \right), \end{aligned} \tag{20}$$

where $C_i^G > 0$ is the cost of producing one unit of power with the controllable generator, $C_i^B > 0$ is a cost associated to the aging of the battery, $C_i^{Pl} > 0$ is the per-unit cost paid by the ESP to prosumer i for changes in its programmable load consumption profile with respect to its original schedule, and $C_i^{Sl} > 0$ is the per-unit cost paid by the ESP associated to how much the non-interruptible load is shifted. In case of multiple devices, the quantities in (20) shall be replaced by the summation over the respective device index. The last term of (20) represents the revenue/cost of the ESP for the mobilized power reserve satisfying the TSO request during the required period (see (18)), properly weighted by the reserve price $p_R > 0$. In particular, the ESP is remunerated when it mobilizes upward power reserve i.e. when $\left(P(t) - \widetilde{P}(t)\right) = \sum_{i=1}^{N} \left(P_i(t) - \widetilde{P}_i(t)\right) > 0$, while it affords a cost when it decreases the output power with respect to the baseline to mobilize downward power reserve, i.e. when $\left(P(t) - \widetilde{P}(t)\right) = \sum_{i=1}^{N} \left(P_i(t) - \widetilde{P}_i(t)\right) < 0$, [13,32]. Notice that the sign of the mobilized reserve in real-time is imposed by the TSO request $\Gamma(t)$ through (18), while (19) imposes that $\left(P(t) - \widetilde{P}(t)\right) = \sum_{i=1}^{N} \left(P_i(t) - \widetilde{P}_i(t)\right) = 0$ when no mFRR services are requested. As common in most reserve markets, the mobilized upward and downward reserves are differently priced [13,35]. To include this aspect, $p_F(t)$ is automatically fixed either to the upward or downward reserve price, based on the sign of the TSO request $\Gamma(t)$, which is known before solving the presented optimization problem. Formally, it follows that

$$p_F(t) = \begin{cases} p_R^{\uparrow}(t), & \text{if } \Gamma(t) \geqslant 0, \\ p_R^{\downarrow}(t), & \text{if } \Gamma(t) < 0, \end{cases} \tag{21}$$

for all $t \in \mathscr{T}$, where $p_R^{\uparrow} > 0$ and $p_R^{\downarrow} > 0$ are the upward and the downward power reserve price, respectively.

The optimal response of the ESP to the TSO demand of flexibility over the time slots from t_0 to t_f can therefore be obtained as the solution of the following optimization program

$$\begin{aligned} &\min_{\substack{P_i^G(t), P_i^B(t), P_i^{B,d}(t), P_i^{B,c}(t), \\ P_i^{Pl}(t), P_i^{Sl}(t),\ P_i(t), \\ \delta_i^G(t), \delta_i^B(t), \delta_i^{Pl}(t), \delta_i^{Sl}(t), \quad t \in \mathscr{T},\ i \in \mathscr{N}}} J(\cdot) \\ &\textit{subject to}: \quad (18) \\ &\left. \begin{aligned} &(6), \ \ t \in \{0, \dots, M-2\}, \\ &\left. \begin{aligned} &(2), (4), (8), (11), (12), \\ &(13a), (14a), (16) \end{aligned} \right\} t \in \mathscr{T} \\ &(13b), (14b), (17), (19) \end{aligned} \right\} i \in \mathscr{N} \end{aligned} \tag{22}$$

where $J(\cdot)$ is expressed in (20) and $\mathscr{T} = \{0, \dots, M-1\}$. Problem (22) involves both continuous ($P_i^G(t)$, $P_i^B(t)$, $P_i^{B,d}(t)$, $P_i^{B,c}(t)$,$P_i^{Pl}(t)$, $P_i^{Sl}(t)$, $P_i(t)$) and discrete ($\delta_i^G(t)$, $\delta_i^B(t)$, $\delta_i^{Pl}(t)$, $\delta_i^{Sl}(t)$) decision variables for each prosumer. Indeed, the cost function and the constraints in the optimization problem (22) are functions of the decision variables either directly or indirectly through(1).

Note that the cost function (20) is additive over the prosumers, so that if the constraint (18) is removed, then problem (22) becomes separable, hence, easier to solve. Constraint (18) is therefore named *coupling constraint* since it couples the prosumers' decisions, whereas the other constraints are referred to as *local*.

Note also that, even though problem (22) is not linear since the cost function in (20) contains absolute values, it can be rewritten as a linear program by adopting an epigraphic reformulation. To this end, let us introduce the auxiliary variables $h_i^B(t)$,$h_i^{Pl}(t)$, and $h_i^{Sl}(t)$ and the following additional local constraints

$$\begin{aligned} &P_i^B(t) - P_i^B(t-1) \leqslant h_i^B(t), \\ &P_i^B(t-1) - P_i^B(t) \leqslant h_i^B(t), \\ &P_i^{Pl}(t) - \widetilde{P}_i^{Pl}(t) \leqslant h_i^{Pl}(t), \\ &\widetilde{P}_i^{Pl}(t) - P_i^{Pl}(t) \leqslant h_i^{Pl}(t), \\ &\sum_{\tau=0}^{M-1} \tau \delta_i^{Sl}(\tau) - \tau_i^{Sl} \leqslant h_i^{Sl}(t), \\ &\tau_i^{Sl} - \sum_{\tau=0}^{M-1} \tau \delta_i^{Sl}(\tau) \leqslant h_i^{Sl}(t), \end{aligned} \tag{23}$$

for all $t \in \mathscr{T}$.

If the cost function is reformulated as

$$J_e(\cdot) = \sum_{i=1}^{N} \sum_{t=t_r+1}^{M-1} C_i^G P_i^G(t) + C_i^B h_i^B(t) + C_i^{Pl} h_i^{Pl}(t) + C_i^{Sl} h_i^{Sl}(t) - p_F(t)\Big(P_i(t) - \widetilde{P}_i(t)\Big), \tag{24}$$

then

$$\begin{array}{c} \mathit{min} \\ P_i^G(t), P_i^B(t), P_i^{B,d}(t), P_i^{B,c}(t), \\ P_i^{Pl}(t), P_i^{Sl}(t), P_i(t), h_i^B(t), h_i^{Pl}(t), h_i^{Sl}(t) \\ \delta_i^G(t), \delta_i^B(t), \delta_i^{Pl}(t), \delta_i^{Sl}(t),\ t \in \mathscr{T},\ i \in \mathscr{N} \end{array} \quad J_e(\cdot) \tag{25}$$

$$\mathit{subject\ to}: \qquad (18)$$

$$\left.\begin{array}{l} (6), \qquad t \in \{0, \ldots, M-2\}, \\ \left.\begin{array}{l} (2), (4), (8), (11), (12), \\ (13a), (14a), (16), (23), \\ (13b), (14b), (17), (19) \end{array}\right\} t \in \mathscr{T} \end{array}\right\} i \in \mathscr{N}$$

is linear and equivalent to (22).

If the decision variables $P_i^G(t)$, $\delta_i^G(t)$, $P_i^B(t)$,$P_i^{B,d}(t)$,$P_i^{B,c}(t)$,$P_i^{Pl}(t)$,$P_i^{Sl}(t)$, $\delta_i^B(t), h_i^B(t), \delta_i^{Pl}(t), h_i^{Pl}(t), \delta_i^{Sl}(t), h_i^{Sl}(t)$, and $P_i(t)$,$t \in \mathscr{T}$, related to prosumer i are collected into a vector x_i, noticing that (24) is also separable across i, then problem (25) can be compactly rewritten as

$$\min_{x_1,\ldots,x_N} \quad \sum_{i=1}^{N} c_i^\top x_i \tag{26}$$

$$\begin{aligned} &\mathit{subject\ to}: \\ &\sum_{i=1}^{N} A_i x_i \leqslant b, \\ &x_i \in X_i, \quad i \in \mathscr{N}, \end{aligned}$$

where X_i is the mixed-integer polyhedral set defined by constraints (2), (4), (6b), (8a), (11), (12), (13), (14), (16), (17), (19), and (23) together with relation (1), while $\sum_{i=1}^{N} A_i x_i \leqslant b$ represents the coupling constraint (18). The compact form (26) for the ESP problem (25) is introduced to facilitate the description of the proposed distributed optimization algorithm. MILP problems of the form (26) are common in many engineering fields, see for instance [36,37], and the development of efficient centralized or distributed algorithms for their solution represents a challenging problem.

3. Proposed distributed strategy

The optimization problem in (26) fits the framework proposed in [34], which provides a scalable distributed strategy for its approximate resolution. The iterative algorithm proposed in [34] exploits dual decomposition to obtain a scalable and privacy preserving solution: agents have to solve in parallel a lower dimensional MILP involving their local decision variables, cost, and constraints, while a central entity is in charge of enforcing the coupling constraint by updating the dual variable based on the tentative solutions of the agents. Constraint tightening is integrated within dual decomposition in order to ensure that a feasible solution to (26) is found after a finite number of iterations.

Duality theory, see e.g., [38], plays a central role in the distributed resolution of multi-agent optimization problems in the form of (26) as it allows to decompose the problem across the agents (the prosumers in this case) by softening the coupling constraint and incorporating it as an additive term in the cost function. More precisely, let $\lambda \geqslant 0$ be a vector of Lagrange multipliers (the dual variable) and let introduce

$$L(x_1, \ldots, x_N, \lambda) = \sum_{i=1}^{N} c_i^\top x_i + \lambda^\top \left(\sum_{i=1}^{N} A_i x_i - b \right), \tag{27}$$

that is the Lagrangian function obtained augmenting the cost function of (26) with a term that penalizes the amount of violation of the coupling constraint weighted by λ. The dual problem of (26) is then given by

$$\max_{\lambda \geqslant 0} -\lambda^\top b + \sum_{i=1}^{N} \underbrace{\min_{x_i \in X_i} \big(c_i^\top + \lambda^\top A_i\big) x_i}_{\varphi_i(\lambda)} \quad . \tag{28}$$

Let us notice that despite (26) is a non-convex program, given the presence of discrete variables, problem (28) is convex as each $\varphi_i(\lambda)$ is a concave function since it is the minimum of affine functions of λ, see also [38, Proposition 5.1.2].

A distributed approach to solve (28) is the well-known dual subgradient algorithm, see [38, Section 6.3], which iterates between the following two steps:

$$x_i(k+1) \in \mathit{argmin}_{x_i \in X_i} \big(c_i^\top + \lambda(k)^\top A_i\big) x_i, \tag{29a}$$

$$\lambda(k+1) = \left[\lambda(k) + \alpha(k) \left(\sum_{i=1}^{N} A_i x_i(k+1) - b \right) \right]_+, \tag{29b}$$

where $\alpha(k)$ is a step-size parameter satisfying $\sum_{k=0}^{\infty} \alpha(k) = \infty$ and $\sum_{k=0}^{\infty} \alpha(k)^2 < \infty$, and $[\cdot]_+$ denotes the projection of its argument onto the non-negative orthant. Update (29a) can be performed in parallel by the agents, while step (29b) has to be performed by a central entity. In the considered setting, the agents are the prosumers and the central entity is the ESP. It is worth noticing that each prosumer i needs to communicate to the ESP only its contribution $A_i x_i(k+1)$ to the coupling constraint (i.e., its power profile $P_i(t)$, see (18)) and is not required to disclose any private information regarding operating costs (coded in c_i) nor device characteristics and limitations (coded in X_i).

Typical choices for $\alpha(k)$ satisfying $\sum_{k=0}^{\infty} \alpha(k) = \infty$ and $\sum_{k=0}^{\infty} \alpha(k)^2 < \infty$ are given by

$$\alpha(k) = \frac{\alpha_1}{(k+1)^{\alpha_2}}, \tag{30}$$

with $\alpha_1 > 0$ and $\alpha_2 \in (0.5, 1]$.

Unfortunately, applying (29) does not provide a way to recover the optimal solution $x_1^\star, \ldots, x_N^\star$ of (26). In absence of integer decision variables, the method in [39, eq. (4.118)] provides a recovery procedure which involves computing a convex combination of the tentative solutions $x_i(k+1)$ explored across all iterations. However, this strategy does not work in presence of integer variables, as averaging will not necessarily return an integer quantity, thus making the recovered solution not necessarily feasible for (26). The approach in [34] overcomes this issue using a different approach. Instead of averaging, it replaces the b vector in (29b) with $b - \rho(k)$, where $\rho(k) > 0$ is a tightening vector having the same dimension of b. The role of $\rho(k)$ is to progressively reduce some components of b to enforce feasibility of the coupling constraint. The update of $\rho(k)$ is based on how much the tentative solutions $x_i(k)$ explored by (29) across iterations affect the coupling constraint (18).

The approach in [34] is guaranteed to return a feasible solution after a finite number of iterations provided that ρ does not grow too much as the algorithm progresses, see [34] for a detailed discussion. This is actually an issue in our set-up, as constraint (18) in (25) poses a limit on the amount ρ can grow, as better explained hereafter.

The tightened version of the coupling constraint, for a generic ρ vector, is given by

$$(1-\varepsilon)\Gamma(t)+\rho_t^{lb}(t)\leqslant \quad P(t)-\widetilde{P}(t) \quad \leqslant(1+\varepsilon)\Gamma(t)-\rho_t^{ub}(t), \tag{31}$$

where $\rho_t^{ub}(t)$ and $\rho_t^{lb}(t)$ are the components of ρ associated with the right and left constraints in (18) respectively, for a given $t\in[t_0,t_f]$. From (31) it is clear that an excessive tightening would quickly lead to infeasibility since

$$\rho_t^{ub}(t)+\rho_t^{lb}(t)\leqslant 2\varepsilon\Gamma(t), \tag{32}$$

$t=t_0,\dots,t_f$, must hold.

In [34], the tightening vector ρ is updated every time there is a change in $x_i(k), i=1,\dots,N$. However, during the first iterations of (29), $\lambda(k)$ will be far from being optimal and the impact of the tentative solutions $x_i(k)$ explored by the prosumers on (18) may be far from optimality, thus leading to an aggressive update strategy for ρ that is likely to violate (32). Ideally, one would instead like to wait for the $\lambda(k)$ to settle to properly assess the impact of $x_i(k)$ on (18) and then update ρ accordingly. Moreover, every time ρ changes, also the limit point of $\lambda(k)$ changes, which suggests to devise a strategy in which one waits for $\lambda(k)$ to settle after each update of ρ before updating ρ again. This intuition lead us to the development of Algorithm 1 which, although inspired by [34], updates the tightening vector ρ more cautiously based on a nested loop strategy. An inner loop is responsible for waiting until the sequence $\lambda(k)$ settles given a fixed ρ, while the outer loop updates ρ after practical convergence of the inner loop, and then the cycle is repeated. In Algorithm 1, the values of ρ generated by the outer loop of Algorithm 1 are indexed by the outer iteration index k_o. At each outer iteration k_o, the inner loop (Steps 6–13) runs, until practical convergence, a modified version of (29), where the b vector in (29b) is replaced by $b-\rho(k_o)$ (see Step 11 in particular), $\rho(k_o)$ being the value of the tightening parameter at the current outer iteration. The sequence generated by the inner loop of Algorithm 1 are indexed by the inner iteration index k_i. They are used to update the tightening coefficient ρ at the end of the outer iteration, initialize the λ sequence for the next run of the inner loop, and then discarded. Note that the same privacy-related considerations for (29) apply also to Algorithm 1.

Algorithm 1. Distributed MILP

```
1.$\lambda(0)=0$
2.$\rho(0)=0$
3.$k_o=0$
4.**repeat**
5.    $k_i=0$
6.    **repeat**
7.        **for** $j=1$ to $N$ **do**
8.            $x_j(k_i+1)\in argmin_{x_j\in X_j}\left(c_j^\top+\lambda(k_i)^\top A_j\right)x_j$
9.        end for
10.       $v(k_i+1)=\sum_{j=1}^N A_j x_j(k_i+1)-b$
11.       $\lambda(k_i+1)=[\lambda(k_i)+\alpha(k_i)(v(k_i)+\rho(k_o))]_+$
12.       $k_i\leftarrow k_i+1$
13.   **until** $\lambda(k_i)$ converges **or** $x_0(k_i),\dots,x_N(k_i)$ is feasible
14.   $\overline{k}_i=argmin_{k>k_i-w}\left\|[v(k)]_+\right\|_\infty$
15.   $\rho(k_o+1)=\rho(k_o)+[v(\overline{k}_i)]_+$
16.   $\lambda(0)=\lambda(k_i)$
17.   $k_o\leftarrow k_o+1$
18.**until** $x_0(k_i),\dots,x_N(k_i)$ is feasible
```

The rationale behind this nested loop strategy is to ignore the tentative primal solutions $x_i(k)$ explored during the transient phase of $\lambda(k)$, which may improperly assess the impact of each prosumer onto the coupling constraint, and focus only on those tentative primal solutions that are computed when $\lambda(k)$ reaches a steady state for the given $\rho(k_o)$. This tentative solution will give a better assessment of the impact of each prosumer to the coupling constraint, which is then used to update ρ between two consecutive outer iterations. This, together with the modified update strategy for ρ described next, ultimately results in a much more cautious update of ρ, as testified in the simulation section. Within the inner loop, the ESP keeps monitoring the violation of the coupling constraints by computing (see Step 10)

$$v(k_i)=\sum_{j=1}^N A_j x_j(k_i)-b, \tag{33}$$

whose r-th component is positive if the r-th joint constraint is violated and is negative or zero otherwise. If, for some k_i, all components of $v(k_i)$ are non-positive, then the algorithm terminates and returns solution $x_0(k_i),\ \dots,\ x_N(k_i)$, which is feasible. Otherwise the inner loop keeps running until $\lambda(k_i)$ converges. When the inner loop has converged, the ESP looks at the latest w values of $v(k)$, and computes, for each $k=k_i-w+1,\dots,k_i$, the amount of maximum violation $\left\|[v(k)]_+\right\|_\infty$ (i.e., the highest among the positive components of $v(k)$), and then selects the inner iteration index corresponding to the lowest $\left\|[v(k)]_+\right\|_\infty$ in Step 14. This translates into selecting the tentative primal solution among the latest w with the least maximum violation of the coupling constraints.

Finally, the update of the tightening coefficient ρ is performed by taking the integral of the violation $[v(\overline{k}_i)]_+$ associated with the least violating tentative primal solution (see Step 15). The rationale behind this consists in trying to increase the tightening vector ρ by a small amount at each outer iteration to avoid being overly conservative.

In the unfortunate event that $\rho(k_o)$ is such that (32) is not satisfied for some $t=t_0,\dots,t_f$, then a feasible solution could not be found. One can then stop the algorithm and restart it with a larger value for the tolerance ε in (18) thus softening the TSO request and possibly providing an approximate yet guaranteed solution to its original request.

Fig. 1 provides a pictorial view of the interaction mechanism involved in Algorithm 1, which is activated by a TSO request to the ESP.

4. Simulation results and discussions

The proposed strategy is tested on a numerical example with $N=50$ prosumers. The 24 h time-horizon is discretized into $M=96$ time slots of duration $\tau_s=15$ minutes each.

In principle, the baseline power profiles of each prosumer i should be obtained using optimization-based strategies explicitly accounting for the provided balancing service, see [10]. In fact, the power baseline is generally defined to maximize the ESP profit considering the energy prices, the internal production costs and the trading of flexibility reserves, e.g. during the day-ahead market stages. Here, to show the effectiveness of the proposed approach independently on the profile generation mechanism, the baseline power profiles (together with devices physical limits and operating costs) are generated at random according to the procedure described in the Appendix. The upward and downward mFRR prices for the activated reserves have been downloaded from the ENTSO-E Transparency Platform [40], considering the profiles that occurred in the French market on March 2, 2021 [41].

Table 2
mFRR requests by the TSO to the ESP coordinating the prosumer pool.

TSO request	$\Gamma(t)$	t_r	t_0	t_f	ϵ
1-st	800 kW	6:45	7:00	9:00	0.05
2-nd	−700 kW	14:45	15:00	17:00	0.05

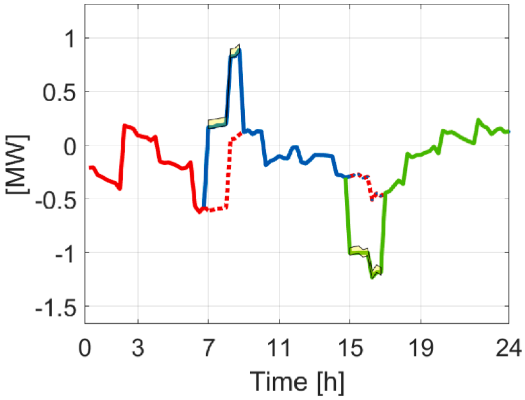


Fig. 2. ESP power profile. Original reference profile (red), profile based on the first request (blue), profile based on the second request (green): scheduled (dotted lines) and actual (solid lines) profiles, and tolerance bounds (yellow band). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

A scenario in which the ESP receives two requests from the TSO is considered: at time 6:45 the TSO asks to increase by 800 kW the amount of power produced by the prosumers pool between 7:00 and 9:00; while at time 14:45 the TSO asks to reduce by 700 kW the amount of power produced between 15:00 and 17:00. Algorithm 1 is therefore executed twice, using in both cases $\alpha(k)$ given by (30), with $\alpha_1 = 0.0035/N$ and $\alpha_2 = 0.51$, and $w = 8$. In the first TSO request, it follows that $t_0 = 28$ and $t_f = 35$ since they correspond to the time intervals 7:00–7:15 and 8:45–9:00, respectively, and $\Gamma(t) = 800$ kW, $t = t_0, \dots, t_f$. In the second TSO request, $t_0 = 60, t_f = 67$, and $\Gamma(t) = -700$ kW, $t = t_0, \dots, t_f$. In both cases the relative tolerance level for the satisfaction of the TSO request is $\varepsilon = 0.05$ and $t_r = t_0 - 1$. For the sake of clarity, the parameters of the TSO requests are also reported in Table 2.

The following color/style code is adopted in the figures. The line color is associated with the time when the power profile has been generated, while line style differentiates between actuated profiles (solid) and scheduled ones (dotted). Red lines refer to reference power profiles, that have been computed before receiving any of the two TSO requests. The red lines are solid until the TSO issues the first request and then become dotted to illustrate the power profiles the prosumers/ESP would have followed if the TSO had not issued any request. Blue lines then refer to power profiles returned by Algorithm 1 in response to the first TSO request. The blue lines are solid until the TSO issues the second request, and become dotted after the second request to report the profiles the prosumers/ESP would have followed if the TSO had not issued a second request. Finally, the green lines refer to power profiles returned by Algorithm 1 in response to the second TSO request. The green lines are all solid because the green profiles are followed until the end of the day, as no other request is issued by the TSO.

Fig. 2 shows how the power profile $P(t), t = 0, \dots, M-1$, changes throughout the day as the two TSO requests are received and processed by the ESP.

The yellow band around the solid lines between 7:00 and 9:00 and 15:00 and 17:00 represent the allowed deviation from the TSO request based on the relative tolerance $\varepsilon = 0.05$, see (18). As can be seen from Fig. 2, for both events, the algorithm is able to reschedule the usage of the prosumers devices so as to meet the TSO request while avoiding the rebound effect, since the declared baseline (red line) is maintained after the satisfaction of the requests.

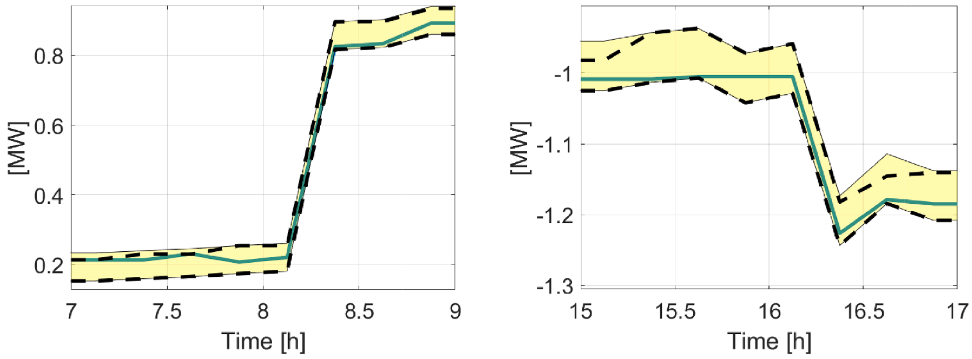
Close-up views of the power profile of the prosumer pool in the time frames 7:00–9:00 and 15:00–17:00 related to the two requests are shown in Figs. 3(a) and 3(b), respectively. Besides the yellow band representing the request satisfaction tolerance bound coded by constraint (18), the tightened tolerance upper and lower bounds (black dashed lines) are also reported, representing constraint (31) for the last value of ρ before the algorithm returned a feasible solution.

The power profiles of one of the prosumers are reported in Fig. 4: its overall power profile $P_i(t)$ in Fig. 4(a), the profile of the generator in Fig. 4(b), the charging/discharging profile of the battery in Fig. 4(c), and the modulation of the load in Fig. 4(d). From Figs. 4(b)-(d) it is evident how the prosumer has changed the schedule of its internal devices to accommodate part of the TSO request while avoiding any rebound effect (the green solid line in Fig. 4(a) matches the reference profile represented by the red dotted line). Specifically, to satisfy the power request between 7:00 and 9:00, the generator and the battery output power are increased, while the controllable load is switched off. During the second TSO request, the generator is switched off, the battery is operated in charging mode, while the load increases its power demand, anticipating the rise with respect to the pre-scheduled profile.

In order to quantify the quality of the feasible solution $\widehat{x}_1, \dots, \widehat{x}_N$ returned by Algorithm 1, the value $\widehat{J}$ of the cost function $J(\cdot)$ in (20), obtained for $\widehat{x}_1, \dots, \widehat{x}_N$, should be compared with the optimal value $J^\star$ of (22), computing the relative optimality gap as follows

$$\Delta J_{\%} = \frac{\widehat{J} - J^\star}{J^\star} \cdot 100. \tag{34}$$

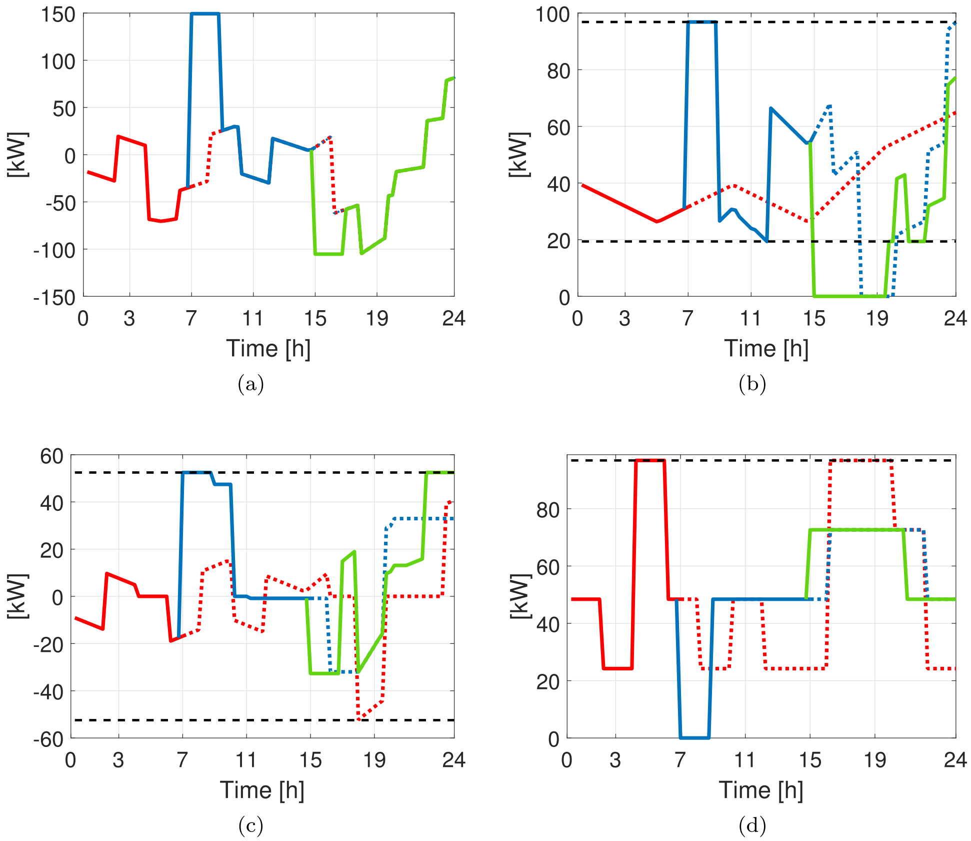
For a large number N of prosumers, problem (22) could not be solved with standard computational resources. In the case-study, for example, out-of-memory issues occurred for large-scale prosumers pool (see Table 4) and it was often not possible to compute $J^\star$ using a server with processor Intel(R) Xeon(R) CPU E5-2630 2.30 GHz and with 64 GB RAM. In fact, the overall problem (22) involves N Boolean variables $\delta_i^G(t)$, N



(a)

(b)

Fig. 3. Close-up view of the power profile of the prosumer pool (blue solid line) in the time frames 7:00–9:00 (a) and 15:00–17:00 (b), together with the tolerance bounds (yellow band) and tightened tolerance bounds (black dashed lines). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) (b) (c) (d)

Fig. 4. Power profiles of prosumer 10: total (a), generator (b), battery (c), programmable load (d). Original reference profile (red), profile based on the first TSO request (blue), profile based on the second TSO request (green): scheduled (dotted lines) and actual (solid lines) profiles, capability limits (dashed lines). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Assessment of the performance of Algorithm 1 when applied to satisfy the two TSO requests with $N = 50$.

TSO request	Outer iterations	Inner iterations	Execution time [sec]	Estimated bound $\overline{\Delta J}_{\%}$
1-st	8	166	29.9	0.15%
2-nd	2	35	2.3	0.25%

Boolean variables $\delta_i^B(t)$, N integer variables $\delta_i^{Pl}(t)$, taking $n_i^{Pl}+1$ discrete values, and N Boolean variables $\delta_i^{Sl}(t)$, for $t \in \{t_r+1, \dots, M-1\}$, thus amounting to a total of $2^{3(M-t_r-1)N} \cdot \prod_{i=1}^{N}(n_i^{Pl}+1)^{(M-t_r-1)}$ possible combinations to be explored in a centralized solution. On the other hand, the proposed distributed approach allows to decompose the overall problem and solve (in parallel!) N sub-problems with only $2^{3(M-t_r-1)} \cdot (n_i^{Pl}+1)^{(M-t_r-1)}$ possible combinations each. This translates into 40^{3400} possible combinations for the centralized problem, and 40^{68} for each prosumer, for the first request; and 40^{1800} and 40^{36}, respectively, for the second request.

While $J^\star$ cannot be computed for large-scale problems, it can be estimated leveraging (28). In fact, by weak duality, it is known that the optimal value $J_D^\star$ of (28) constitutes a lower bound for $J^\star$ (see [38, Proposition 5.1.3]). Therefore, since $\widehat{x}_1, \dots, \widehat{x}_N$ is feasible for (22), it follows that

$$J_D^\star \leqslant J^\star \leqslant \widehat{J}, \tag{35}$$

and the relative optimality gap can therefore be estimated as follows

$$\Delta J_{\%} = \frac{\widehat{J} - J^\star}{J^\star} \cdot 100 \leqslant \frac{\widehat{J} - J_D^\star}{J_D^\star} \cdot 100 = \overline{\Delta J}_{\%}, \tag{36}$$

where the value of $J_D^\star$ can be easily computed iterating (29) until convergence.

In Table 3, the estimated relative optimality gap $\overline{\Delta J}_{\%}$ for the two TSO requests is reported, along with the number of outer iterations performed, which also corresponds to the number of times the value of ρ has been updated, the total number of inner iterations, which corresponds to the number of times each agent has performed Step 8, and the time elapsed before Algorithm 1 returned a feasible solution divided by the number N of prosumers. The latter being a good indicator of how long the algorithm will take in practice as the time elapsed performing Steps 10–15 is negligible with respect to the time needed by each prosumer to perform (in parallel) Step 8.

Remarkably, for the two cases reported in Table 3, the bound on the relative optimality gap is lower than 1%, showing the effectiveness of the approach in returning a feasible solution with close-to-optimality performance. Note also how the first request exhibits a much higher execution time with respect to the second one. This is due not only to the larger number of iterations, but also to the longer time horizon the prosumer have to consider for the rescheduling phase after the first TSO request, with respect to the second one. Note also that execution times are of the order of seconds, which is significantly less then the 15 min time interval available to the ESP for implementing the TSO request. Finally, it is worth noticing that, in both cases, the number of outer iterations is greater than zero (i.e. the final ρ is different from zero). This witnesses that the standard dual subgradient method, executed in the first outer iteration of Algorithm 1 (i.e. Steps 6–13 with $\rho(0) = 0$), is not

Table 4
Performance comparison between the distributed Algorithm 1 and the centralized solution of (25) as a function of the number N of prosumers in the pool. (*) The solver stopped as the available RAM memory (64 GB) was saturated, i.e. the obtained solution is not optimal.

	Algorithm 1 (Distributed)			Centralized Algorithm	Relative optimality gap	
N	Outer iterations	Inner iterations	Execution time [sec]	Execution time [sec]	Real $\Delta J_{\%}$	Estimated bound $\overline{\Delta J}_{\%}$
30	5	110	7.9	102.5	0.26%	0.29%
40	2	106	10.6	156.7	0.23%	0.28%
50	3	87	14.0	$2{\cdot}10^4$ (*)	0.24% (*)	0.26%
60	7	172	26.7	$3.6{\cdot}10^4$ (*)	0.22% (*)	0.22%
70	3	127	8.1	$2.1{\cdot}10^4$ (*)	0.06% (*)	0.07%
80	12	231	15.9	$1.5{\cdot}10^4$ (*)	0.06% (*)	0.10%
1000	10	167	10.0	$1.5{\cdot}10^4$ (*)	0.02% (*)	0.02%

able to find a feasible solution.

For comparison purposes, the original algorithm in [34] was also tested, but it resulted in condition (32) becoming violated after the first few iterations.

The proposed distributed strategy has also been tested for a growing number N of prosumers in the pool, to assess its scalability properties. Moreover, the test-cases have been also solved through a centralized algorithm, directly solving (25), in order to highlight the advantages of the proposed approach.

A TSO request of $\Gamma(t) = 18N$ kW between 15:00 ($t_0 = 60$) and 17:00 ($t_f = 67$) has been assumed for each test case, with all the other parameters set as previously discussed.

The performance of Algorithm 1 is evaluated based on the same indicators of Table 3 for different values of N and the results are reported in Table 4. The different test-cases have been also addressed through a centralized algorithm, solving directly (25). As it appears from the table, for each N, the proposed distributed approach is able to find a feasible solution with close-to-optimal performance ($\Delta J_{\%} < 1\%$) in a handful of iterations and in less than 30 s. On the other hand, the centralized algorithm always involves larger computational times and it is not able to find the optimal solution for most of the test-cases, as the available RAM memory (64 GB) was saturated by the solver for $N \geqslant 50$. Note that for this test-cases problem (25) involves $504N$ variables: $360N$ continuous, $108N$ boolean, and $36N$ integer assuming 5 possible values. The total number of possible combinations is thus 40^{36N}, which eventually lead to an intractable problem as N grow, when addressed centrally.

5. Conclusion

In this paper, the provision of balancing services to the TSO is addressed, and, in particular, the manual Frequency Restoration Reserve (mFRR) service, through an aggregation of prosumers instead of a single large industrial one. A main issue in such a context is to devise a suitable strategy for the ESP to coordinate a possibly large number of prosumers and distribute the effort so as to satisfy the TSO requests of power flexibility.

A framework is proposed that relies on a MILP formulation of the problem and on its solution via distributed computations. The resulting approach is scalable since it decomposes the overall problem into subproblems of the size of the decision variables of the prosumers, provides a privacy-preserving solution since the prosumers do not need to share with the ESP their actuation capabilities and operational costs, and can handle integer decision variables.

The approach should be extended to the case when non-controllable loads or generators are present, possibly adopting a stochastic framework and a different concept of balancing service, where the required power variation is met with a given (high) probability. Theoretical analysis of the convergence and the feasibility properties of the proposed algorithm requires additional research effort and is left for future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The work of Alessio La Bella has been financed by the Research Fund for the Italian Electrical System in compliance with the Decree of Minister of Economic Development April 16, 2018.

Appendix A. Appendix: Generation of the simulation set-up

Consider the generic prosumer i. As for the programmable load, $\overline{P}_i^{Pl}$ is extracted at random according to the uniform distribution over the interval $[10,100]$ kW and the reference profile $\widetilde{P}_i^{Pl}(t)$ is set equal to a piecewise constant function with intervals of 2-h duration each and amount of power per interval computed according to (12a) with $\delta_i^{Pl}(t)$ extracted uniformly at random in $\{1,\dots,n_i^{Pl}\}$ with $n_i^{Pl} = 4$. The overall amount of energy required by the load and appearing in (13b) is then given by $E_i^{Pl} = \sum_{t=0}^{M-1}\tau_s\widetilde{P}_i^{Pl}(t)$. The load flexibility limits are set to the entire optimization horizon, i.e., $t_i^{Pl,0} = t_0$ and $t_i^{Pl,f} = M-1$. All loads are assumed to be interruptible.

The generators are set such that $\overline{P}_i^G = \overline{P}_i^{Pl}, \underline{P}_i^G = 0.2\overline{P}_i^G$, and the reference profile as a piecewise linear function with hinging points every 4 h with values given by the average power consumption of the load scaled by a factor chosen uniformly at random from the set $\{0.5, 0.75, 1, 1.25, 1.5\}$. Minimum up and down times for the generators were assumed to be equal and set to $T_i^{G,u} = T_i^{G,d} = 8$.

As for the battery capacity limits, it has been set that $\overline{S}_i = \frac{1}{4}E_i^{Pl}$ so that the battery is able to provide 25% of the energy needed by the load, $\underline{S}_i = 0.1\overline{S}_i$, whereas its initial content $S_i(0)$ is extracted at random from $\left[\underline{S}_i, \overline{S}_i\right]$ according to a uniform distribution. The battery power bounds, i.e., $\overline{P}_i^B$ and $\underline{P}_i^B$, are defined as $\overline{P}_i^B = -\underline{P}_i^B = \overline{S}_i/\left(\frac{\tau_s M}{4}\right)$ so that the battery can be fully charged/discharged in a quarter of a day, i.e. 6 h. Battery discharging/charging coefficients are set to $\zeta_i^d = 1/0.95$ and $\zeta_i^c = 0.95$ respectively. The battery reference power profile $\widetilde{P}_i^B(t)$ is defined so as to match $\widetilde{P}_i^G - \widetilde{P}_i^{Pl}$ compatibly with its rate and capacity constraints.

Finally, the unitary costs C_i^{Pl}, C_i^G, and C_i^B appearing in the cost function (20) are extracted at random according to a uniform distribution over the interval $[0, 35]$ euro/kW for C_i^{Pl}, and over the interval $[0, 1]$ euro/kW for both C_i^G and C_i^B.

References

[1] Hirth L, Ziegenhagen I. Balancing power and variable renewables: Three links. Renew. Sustain. Energy Rev. 2015;50:1035–51.

[2] Samad T, Koch E, Stluka P. Automated demand response for smart buildings and microgrids: The state of the practice and research challenges. Proc. IEEE 2016;104 (4):726–44.

[3] La Bella A, Farina M, Sandroni C, Scattolini R. Microgrids aggregation management providing ancillary services. 2018 European Control Conference (ECC), IEEE 2018: 1136–41. https://doi.org/10.23919/ECC.2018.8550401.

[4] Yuen C, Oudalov A, Timbus A. The provision of frequency control reserves from multiple microgrids. IEEE Trans. Industr. Electron. 2010;58(1):173–83.

[5] Carreiro AM, Jorge HM, Antunes CH. Energy management systems aggregators: A literature survey. Renew. Sustain. Energy Rev. 2017;73:1160–72.

[6] Koliou E, Eid C, Chaves-Ávila JP, Hakvoort RA. Demand response in liberalized electricity markets: Analysis of aggregated load participation in the german balancing mechanism. Energy 2014;71:245–54.

[7] Müller FL, Szabó J, Sundström O, Lygeros J. Aggregation and disaggregation of energetic flexibility from distributed energy resources. IEEE Transactions on Smart Grid 2019;10(2):1205–14. https://doi.org/10.1109/TSG.2017.2761439.

[8] Di Somma M, Graditi G, Siano P. Optimal bidding strategy for a DER aggregator in the day-ahead market in the presence of demand flexibility. IEEE Trans. Industr. Electron. 2019;66(2):1509–19. https://doi.org/10.1109/TIE.2018.2829677.

[9] Diekerhof M, Schwarz S, Martin F, Monti A. Distributed optimization for scheduling electrical demand in complex city districts. IEEE Syst. J. 2017;12(4): 3226–37.

[10] La Bella A, Farina M, Sandroni C, Scattolini R. Design of aggregators for the day-ahead management of microgrids providing active and reactive power services. IEEE Transactions on Control Systems Technology 2020;28(6):2616–24. https://doi.org/10.1109/TCST.2019.2939992.

[11] Poplavskaya K, de Vries L. Distributed energy resources and the organized balancing market: A symbiosis yet? Case of three european balancing markets, Energy policy 2019;126:264–76.

[12] Gomez T, Herrero I, Rodilla P, Escobar R, Lanza S, de la Fuente I, Llorens ML, Junco P. European union electricity markets: Current practice and future view. IEEE Power Energ. Mag. 2019;17(1):20–31. https://doi.org/10.1109/MPE.2018.2871739.

[13] Iria JP, Soares FJ, Matos MA. Trading small prosumers flexibility in the energy and tertiary reserve markets. IEEE Transactions on Smart Grid 2018;10(3):2371–82.

[14] Khodadadi A, Herre L, Shinde P, Eriksson R, Söder L, Amelin M. Nordic balancing markets: Overview of market rules. In: 2020 17th International Conference on the European Energy Market (EEM), IEEE; 2020. p. 1–6.

[15] TERNA, Italian grid code, https://www.terna.it/en/electric-system/grid-codes/italian-grid-code, accessed: 2021-03-28.

[16] Commission Regulation (EU) 2017/2195 of 23 November 2017 establishing a guideline on electricity balancing, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32017R2195.

[17] F. Oyj, Terms and conditions for providers of manual frequency restoration reserves (mFRR), https://www.fingrid.fi/globalassets/dokumentit/en/electricity-market/reserves/reservitoimittajien-mfrr-ehdot-ja-edellytykset_en.PDF, accessed: 2021-03-28.

[18] Kolenc M, Nemček P, Gutschi C, Suljanović N, Zajc M. Performance evaluation of a virtual power plant communication system providing ancillary services. Electric Power Systems Research 2017;149:46–54.

[19] ENTSO-E, Survey on ancillary services procurement, balancing market design 2017, https://eepublicdownloads.entsoe.eu/clean-documents/Publications/Market%20Committee%20publications/ENTSO-E_AS_survey_2017.pdf, accessed: 2021-03-28.

[20] La Bella A, Bonassi F, Sandroni C, Fagiano L, Scattolini R. A hierarchical approach for balancing service provision by microgrids aggregators. IFAC-PapersOnLine 2020;53(2):12390–935. https://doi.org/10.1016/j.ifacol.2020.12.2126.

[21] Vrettos E, Oldewurtel F, Andersson G. Robust energy-constrained frequency reserves from aggregations of commercial buildings. IEEE Trans. Power Syst. 2016; 31(6):4272–85.

[22] Correa-Florez CA, Michiorri A, Kariniotakis G. Optimal participation of residential aggregators in energy and local flexibility markets. IEEE Transactions on Smart Grid 2019;11(2):1644–56.

[23] Logenthiran T, Srinivasan D, Shun TZ. Demand side management in smart grid using heuristic optimization. IEEE Transactions on Smart Grid 2012;3(3):1244–52.

[24] Pedrasa MAA, Spooner TD, MacGill IF. Coordinated scheduling of residential distributed energy resources to optimize smart home energy services. IEEE Transactions on Smart Grid 2010;1(2):134–43.

[25] Anjos MF, Lodi A, Tanneau M. A decentralized framework for the optimal coordination of distributed energy resources. IEEE Trans. Power Syst. 2018;34(1): 349–59.

[26] Chavali P, Yang P, Nehorai A. A distributed algorithm of appliance scheduling for home energy management system. IEEE Transactions on Smart Grid 2014;5(1): 282–90.

[27] Kim S-J, Giannakis GB. Scalable and robust demand response with mixed-integer constraints. IEEE Transactions on Smart Grid 2013;4(4):2089–99.

[28] Mhanna S, Chapman AC, Verbič G. A distributed algorithm for demand response with mixed-integer variables. IEEE Transactions on Smart Grid 2016;7(3):1754–5.

[29] P. McNamara, S. McLoone, Hierarchical demand response using dantzig-wolfe decomposition, in: IEEE PES ISGT Europe 2013, IEEE, 2013, pp. 1–5.

[30] Amini MH, McNamara P, Weng P, Karabasoglu O, Xu Y. Hierarchical electric vehicle charging aggregator strategy using dantzig-wolfe decomposition. IEEE Design & Test 2017;35(6):25–36.

[31] Zhou X, Dall'Anese E, Chen L. Online stochastic optimization of networked distributed energy resources. IEEE Trans. Autom. Control 2019;65(6):2387–401.

[32] Firoozi H, Khajeh H, Laaksonen H. Optimized operation of local energy community providing frequency restoration reserve. IEEE Access 2020;8:180558–75.

[33] Iria J, Soares F. Real-time provision of multiple electricity market products by an aggregator of prosumers. Appl. Energy 2019;255:113792.

[34] Falsone A, Margellos K, Prandini M. A decentralized approach to multi-agent MILPs': finite-time feasibility and performance guarantees. Automatica 2019;103: 141–50.

[35] Bessa RJ, Matos MA. Optimization models for EV aggregator participation in a manual reserve market. IEEE Trans. Power Syst. 2013;28(3):3085–95.

[36] Liu C, Li K, Li K. Minimal cost server configuration for meeting time-varying resource demands in cloud centers. IEEE Trans. Parallel Distrib. Syst. 2018;29(11): 2503–13.

[37] Liu C, Li K, Xu C, Li K. Strategy configurations of multiple users competition for cloud service reservation. IEEE Trans. Parallel Distrib. Syst. 2015;27(2):508–20.

[38] Bertsekas DP. Nonlinear programming. Journal of the Operational Research Society 1997;48(3). 334–334.

[39] Shor NZ. Minimization methods for non-differentiable functions. Springer; 1985.

[40] Hirth L, Mühlenpfordt J, Bulkeley M. The ENTSO-E Transparency Platform–A review of Europe's most ambitious electricity data platform. Applied energy 2018; 225:1054–67.

[41] ENTSO-E Transparency Platform, https://transparency.entsoe.eu/, accessed: 2021-03-03.