Restricted Neighborhood Communication Improves Decentralized Demand-Side Load Management

Alessandro Giusti, Matteo Salani, Gianni A. Di Caro, Andrea E. Rizzoli, and Luca M. Gambardella
Dalle Molle Institute for Artificial Intelligence (IDSIA), USI and SUPSI — Lugano, Switzerland
{alessandrog,matteo.salani,gianni,andrea,luca}@idsia.ch

Abstract—We address demand-side management of dispatchable loads in a residential microgrid by means of decentralized controllers deployed in each household. Controllers simultaneously optimize two possibly conflicting objectives: minimization of energy costs for the end user (considering a known, time-dependent tariff) and stabilization of the aggregate load profile (load flattening). The former objective can be optimized independently by each controller. On the other hand, the latter could benefit from a communication infrastructure that allows the controllers to explicitly exchange information and coordinate.

To study how different levels of communication pervasiveness affect system performance, we developed a realistic micro-simulation environment accounting for the behavior of residents, dispatchable and non-dispatchable household loads, and the effects on the distribution network. We considered a generic model of communication among household controllers, not tied to any specific technology, and based on the partitioning of the households in a number of groups (neighborhoods). Controllers within the same neighborhood enjoy full connectivity, but cannot interact with controllers outside of their neighborhood. Through extensive simulation experiments, we observed that even communication neighborhoods constituted by as few as 3-4 households are sufficient to effectively stabilize the aggregate network load profile, with minimal bandwidth consumption. Increasing the neighborhood size leads to comparatively negligible performance improvements. We conclude that effective load flattening can be achieved with minimal requirements of communication infrastructure and transmitted information.

I. INTRODUCTION

Nowadays, modern technologies for distributed generation and energy storage are available, which open new perspectives for a smarter use of energy. In particular, controllable production and consumption units are key pieces to form the so-called Smart Grid.

Smart Grid technologies enable the optimized management of the supply side [1], [2] and of the demand side [3], [4]. To robustly achieve these goals, a Smart Grid is expected to be equipped with reliable two-way communication systems. Although the debate on the most adequate communication technologies is still open [5], investments on communication systems enabling Smart Grid technologies are estimated to be between 85 and 192 billions of dollars for USA, by 2030 [6]. Despite the significant potential savings [7], it is still not clear who should sustain these development costs.

Moving from expectations to real implementations, communication approaches for the Smart Grid rise many concerns. For example, in the context of the Advanced Metering Infrastructure, “The brute force solution of polling all the sensors can become the true bottleneck for the sheer problem of collecting all the data in a timely way” moreover, “Delivering messages to Smart Grid terminals through many relays will produce a broadcast storm if protocols to support this function are not designed judiciously” (quoted from [5]).

For the above reasons, it is hard to believe that a fully centralized control approach is going to be at the core of future Smart Grid implementations. Decentralized control approaches, on the contrary, may allow different protocols and standards to coexist, and most likely would result in less expensive communication infrastructure investments.

In this context, our study investigates the impact of different levels of communication pervasiveness. The study builds on the micro-simulation and optimization framework developed in [8] for fully decentralized Demand Side Management (DSM) of household loads in a distribution network [9], [10]. Each household is equipped with a DSM controller which aims to achieve two objectives: a), minimization of energy costs for the end user, and b), stabilization of the load profile (load flattening) at the MV/LV transformer. To optimize the latter objective, different controllers coordinate through a given communication infrastructure, whose minimal requirements we aim to investigate.

To do so, we partition households in small groups (neighborhoods), which e.g. may depend on the topology of the LV distribution network. We simulate a communication infrastructure that only allows controllers in the same neighborhood to communicate. Periodically, each controller schedules demand-side activities accounting for the load foreseen in the neighborhood, and communicates the resulting plan. Neighbor controllers that receive the plan, take it into account during their subsequent optimization runs.

The main contribution of the paper is the resulting asynchronous communication and distributed coordination scheme. Using a test case of 120 households, we provide extensive simulation results for different realistic scenarios. We show that the use of even very small communication neighborhoods, which requires minimal bandwidth and infrastructure, allows DSM controllers to effectively coordinate and stabilize the aggregate load profile. Larger communication neighborhoods lead to comparatively negligible improvements.

The paper is organized as follows. In Section II we provide the modeling details for the simulation framework, the optimization algorithm and communication schemes. Section III reports simulation results for multiple test scenarios. Finally, in Section IV we summarize our findings.
II. Model

We consider a smart grid model based on a micro-simulation of households (Section II-A) including: energy buffers, non-preemptible and non-preemptible load jobs, plug-in electric vehicles, as well as residents and their behavior. Households are connected to a section of the distribution network (Section II-B). Household loads are scheduled by DSM controllers, based on the lexicographic multi-objective optimization algorithm introduced in [8], which is summarized in Sections II-C and II-D. The different controllers interact, as described in Section II-E, by means of a communication infrastructure, detailed in Section II-F. The state of each entity is simulated in discrete time steps of 10 seconds.

A. Simulation of household loads

We consider a typical residential area, and model the main components drawing energy from the grid. Appliances within each household are simulated accounting for their physical state and operational characteristics. We consider four classes of appliances: energy buffers, non-preemptible loads, batteries capable of bidirectional charging, and electric vehicle chargers (possibly bidirectional).

Appliance classes

Energy buffers model appliances such as water heaters, air conditioning/heating systems, and fridges. Electrical energy is used by the appliances to keep the system in a defined operational state (e.g., a given temperature range). Energy buffers account to the largest portion of household loads, and their control has been discussed in several works [11], [12]. The characteristics of an energy buffer are: heat/energy capacity (e.g., depending on the capacity of the heater tank) and a self-discharge rate (e.g., thermal dispersion). Energy buffers are kept in the operating range by switching on and off the heating/freezing elements – whose non-ideal thermal efficiencies are also accounted for. The simulation also considers external factors that influence appliance’s state (e.g., a household resident using hot water at specific times, causing cold water to mix with hot water in the heater). For example, simulating a water heater during a given 10-seconds time step requires to account for its initial internal temperature, the ambient temperature, and the intake of cold water – which in turn depends on the resident behavior. If the heating element is on (which is determined by the control strategy, as described in the following), the resulting energy consumption is recorded, and the consequent temperature increase is computed. This determines the initial state for the following time step.

Batteries for energy storage behave similarly to energy buffers. We simulate the evolution of their state of charge (SOC), accounting for the self-discharge rate, maximum charge, maximum charging/discharging power, and charging/discharging efficiency. Energy stored in batteries can be discharged when required to stabilize network state.

Electric vehicle (EV) chargers are simulated by taking into account the properties of batteries mentioned above, as well as differentiated fast and slow charging modes according to battery requirements. Simulated residents define release and due dates for the electric vehicles, which correspond, respectively, to the times when the vehicle is plugged in after being used, and plugged out before use. After use, some fraction of the battery charge is depleted, according to the simulated behavior of the resident. While the vehicle is plugged, battery charging (and, possibly, discharging) must be handled such that vehicle’s battery is charged over a given threshold (90% in our tests) before the due date. Smart control of EV charging is subject to significant attention by researchers [13], due to its easy controllability and expected increase in EV penetration rates.

Non-preemptible loads include washing machines, dishwashers, and all appliances operating according to a pre-determined working program. Such appliances are normally idle. When started, they draw a pre-defined electrical load profile (for example, a high load during the initial phases of a washing machine run, for heating water, followed by lower load during mechanical washing phases). We simulate a single fixed program for each different appliance. Once started, the appliance only stops when it reaches the end of the program. Household residents define jobs to be executed. Each job is characterized by a release date (before which the program cannot be started) and a due date (before which the program must be completed). The appropriate starting time for the program must meet such strict requirements.

Non-controllable loads model appliances like TV sets, house lighting, and kitchen appliances, which are driven uniquely by simulated residents. Therefore, they are not subject to DSM controllers.

Appliance control approaches

For each controllable appliance, we consider two different approaches. a) A baseline control approach mirrors the simple control policies commonly implemented in the devices, which do not account for energy cost or aggregate load stability, but only ensure that operational constraints are met (e.g., a water heater keeps water temperature within a given range). b) An optimized control approach, which selects the best possible plan to meet given objectives while satisfying the operational requirements of appliances.

In the long run, both approaches draw approximately the same amount of energy. However, the latter allows loads to be shifted in time in such a way that the control objectives (as introduced in Section II-C) are optimized.

In the baseline case, energy buffers are controlled by means of thermostats with hysteresis, driven by a set point temperature and deadband, defining lower and upper thresholds. Temperature is maintained within the deadband. Optimization algorithms obtain the same goal by means of optimized schedules. For instance, if the current energy price is low but it is known to increase, an algorithm optimizing the energy price would turn on the heater in advance, even when the temperature is still above the lower threshold, with the goal of reaching the upper threshold just before the energy price increases. For non-preemptible loads, the baseline approach starts the appliance at the latest time which guarantees that the program is finished by the due date. Instead, control algorithms can select the starting time to optimize over periods of heavy
load, while still meeting the operational requirements. Bidirectional batteries are left idle in the baseline case, while they are exploited by control algorithms to meet their optimization objectives. Electric vehicle chargers, in the baseline case, start charging the vehicle battery when the vehicle is plugged in, and only stop when the required charging threshold is reached. Instead, optimized control algorithms may temporarily stop the charging process – or even partially discharge a bidirectional battery – if this still allows to meet the resident-defined deadline.

**Household residents**
We simulate household residents as stochastic finite state machines; their behavior is driven by probabilistic models, which, every day, instantiate a different variation of a baseline routine, which is different for each resident, and accounts for the time of day. Residents interact with appliances in several ways: they use electric vehicles during working hours, and set feasible due dates when plugging for recharge; residents use hot water at reasonable hours, which results in mixing cold water in the water heater; they program non-preemptible load jobs at regular intervals for dishwashers and washing machines and define feasible due dates. Finally, residents trigger non-controllable loads at appropriate times, such as electric cooking appliances, lighting and entertainment appliances.

**B. Simulation of multiple households**
In our tests, a single simulation covers a period of 10 days and encompasses a total of 120 households, including their residents and appliances. All such households are connected to a portion of the distribution network downstream a single MV-LV transformer.

After a simulation is completed, we can observe the aggregate load profile at the transformer, given by the combined loads of all households (discounted for local generation, e.g., energy deriving from the discharge of bidirectional EV batteries). The volatility of such profile is measured (see Section III) to quantify how well the controllers could optimize the load stability. Moreover, for each household we compute the total price paid for energy, by considering a time-dependent pricing signal, which several households under the same transformer need to schedule the charging of a number of electric vehicles. If not changed and used to optimize future decisions with the goal of flattening the aggregate load profile.

**C. Control and Objectives**
Each controller is based on a mathematical model (see [8] for details), which accounts for the operational constraints of all controllable devices in the household\(^1\); for example, it states that a water heater’s temperature must always stay within a given temperature range, or that a nonpreemptible appliance’s program must be finished before a given deadline previously defined by the resident. The model encompasses variables encoding the current and future states for all such appliances, up to a temporal horizon of 24 hours, which is divided in a set \(T\) of 96 time slots, each with the same duration \(\Delta t = 15\text{min}\). After the model is solved, one can determine the desired state of each appliance for each time slot. The mathematical model is solved according to two possibly contrasting objectives: minimization of the energy costs for the end user, and minimization of the fluctuations in the aggregate network load.

For a given household, energy costs are optimized by minimizing the objective function \(z_c = \sum_{t \in T} c^t P^t \Delta t\), where \(P^t\) corresponds to the total power drawn by the household at timeslot \(t\), and \(c^t\) the cost of energy at \(t\). Optimizing such objective for each household results in the optimization of the final measured performance measure, i.e., the average energy cost for all set of households.

Minimizing fluctuations in the aggregate network load is not as straightforward. One trivial strategy consists in minimizing, in each household, the objective function \(z_h = \sum_{t \in \Delta t} (P^t \Delta t)^m\). Note that, given a total amount of energy used in \(T\), \(z_h\) is minimal when the load profile is as flat as possible. On the contrary, \(z_h\) is large if the same amount of energy is drawn following a fluctuating load profile. In this case, each household attempts to flatten its own load profile as much as possible, which does not necessarily imply that the load profile at the MV/LV transformer will be flat. Consider the case in which several households under the same transformer need to schedule the charging of a number of electric vehicles. If not explicitly coordinating with each other, the controllers may schedule overlapping charging periods, thus causing peaks in the aggregate load curve which could have been avoided.

To overcome this issue, we introduce a network load signal, modeled as a function \(S\) of time. \(S^t\) predicts the overall aggregate load for each time step \(t\), and acts as a penalty function, in that a controller should avoid placing a load at time steps where \(S^t\) is large. Assuming that the load of a single household is negligible with respect to the aggregate load, the load stability objective can be stated as follows:

\[
z_s = \frac{\sum_{t \in T} S^t P^t \Delta t}{T}.
\]

The result of using such an objective is that the controller would shift its load away from peaks, moving it towards

\(^1\)An appendix with the full formulation of the mathematical model is available as supplementary material at http://www.idsi.ch/~giusti/tsg
periods with lower aggregate load (i.e., low values for \( S^t \)). \( S \) is computed by means of communication with other controllers in the same neighborhood, as described in Section II-E.

D. Multi-objective optimization

Since control actions for DSM are implemented at residential level on end user’s appliances [15], smart grid control is expected to be consumer-centric [16]. Therefore, we consider the energy costs as being the primary objective, as we also did in [8]. Objectives are ordered in a lexicographic way [17], [18], with \( z_c \) set as primary objective and \( z_s \) as secondary objective.

In practice, we first perform an optimization run by disregarding the network load \( z_s \) objective value; let \( z_c^* \) denote the resulting optimal value for \( z_c \) – i.e. the minimum achievable energy cost for the user:

\[
 z_c^* = \min z_c = \sum_{t \in T} e^t P^t \Delta t . \tag{2}
\]

Once \( z_c^* \) is known, the optimization is run again; we now optimize the secondary objective function (3) subject to the additional constraint (4):

\[
 \min z_s = \sum_{t \in T} S^t P^t \Delta t \tag{3} \\
 z_c \leq (1 + \eta) z_c^* . \tag{4}
\]

Constraint (4) bounds the deviation from the optimum of the primary objective – i.e., forces energy cost to remain close to \( z_c^* \) – while optimizing the secondary objective (3). The allowed deviation is controlled by parameter \( \eta \), which is set to \( \eta = 0 \) in our experiments. As a result, controllers only shift loads to optimize \( z_s \) when this does not negatively affect the energy cost. Since we perform experiments using the swiss bi-level energy tariff [8], algorithms can exploit the presence of large periods with flat tariffs to optimize \( z_s \). More dynamic pricing profiles lead to different results. The interested reader can find in [8] more comprehensive experiments varying the value of \( \eta \) and for different energy cost scenarios.

Both the simulation environment and the optimization algorithm are implemented in Java. The optimization subproblems are solved by means of mathematical programming. The model adopted in this case has been presented in [8]. It is a Mixed Integer Linear Program. Subproblems are solved using GLPK [19], an open source solver for linear programs, via a standard Branch & Cut algorithm which ensures the convergence to a global optimum.

E. Distributed optimization

As previously mentioned, optimizing the \( z_c \) objective requires no explicit communication, assuming that the energy price profile is known. On the contrary, optimizing \( z_s \) requires to compute the penalty function \( S \), which estimates current and future values for the aggregate network load.

In case no communication infrastructure exists between households, such value cannot be computed, and the network load objective is not considered by controllers.

When a bidirectional communication infrastructure is in place, we define the concept of neighborhood as the set of households able to communicate among each other. We define the full neighborhood case as the scenario in which all households connected to the same MV-LV transformer can communicate among them. We also study the scenario in which the households under a transformer are partitioned in a set of smaller, distinct neighborhoods, composed by as few as two households.

Every household computes its own forecast load for the next 24 hours, and broadcasts this information to its neighbors. The forecast load is computed by the household accounting for the contribution of both for its controllable and non-controllable appliances. In particular, future loads of controllable appliances can be precisely known once a schedule for them has been computed (as a result of the optimization procedure introduced above). Future loads of non-controllable appliances are predicted by considering the average load of such appliances as measured during the previous 3 months for the same time of the day and day of week. More sophisticated forecasting mechanisms can be implemented, which we leave for future work.

Every household receives information about the forecast load profile of each of its neighbors. Such profiles are summed to yield an aggregate forecast profile \( S \), which is used as an estimate of the transformer load in the minimization of the \( z_s \) objective. As a result, a household’s loads tend to be shifted towards the time periods that present lower aggregate loads, according to what neighbors households have advertised.

Each controller repeatedly solves the optimization problem described in Section II-C. An optimization run is triggered at random time intervals, uniformly distributed in the range from 0 to 60 minutes. This ensures that the households are not synchronized. Along with this controlled activation, the optimization process is also triggered every time a new load job is requested by the residents (e.g., when an EV is plugged).

Once the optimization problem is solved, the forecast for the household load profile for the next 24 hours is recomputed and broadcast to the neighborhood. Neighboring controllers can therefore take into account this updated load forecast during their next optimization runs.

Because households are not synchronized, we can assume that, in general, optimization runs in the different households are not executed at the same time. This means that, when the optimization is run, a household has the most up-to-date information available from each one of its neighbors. In this case, convergence of the distributed algorithm can be mathematically proven for neighborhood containing 2 households. In this case, from the definition of \( z_s \) in (3) follows that the objective value is the same for both households; if external conditions are constant, every time a household optimizes its objective function, it will decrease the \( z_s \) value for both households, which implies convergence. In case of larger neighborhoods, convergence can not be mathematically proven; while it is possible that a different schedule is generated every time the algorithm runs in a household (i.e. on average twice per hour), this bears no negative consequences since all such solutions share a low value of the objective function. Consider the case...
in which one controller shifts a load in time to reduce the $z_s$ objective value. From the definition of $z_s$ in (3) follows that, because of this action, the forecast aggregate load profile in the neighborhood is made flatter than it was before the shift happened. In fact, in simulated experiments no instability is observed for any neighborhood size (Section III).

F. Communication infrastructure, requirements and protocol

Since communication occurs exclusively among DSM controllers in the same neighborhood, no strict technological requirements need to be imposed on the characteristics of the communication infrastructure. Communication losses among controllers would result in a smooth degradation of system’s load flattening performance: in the worst case, if all controllers lost communication abilities, the system would behave like in the baseline case shown in Section III. In addition, since the approach is fully decentralized and does not require a central coordinator, the choice of communication infrastructure is flexible and can potentially be different for different neighborhoods.

The communication protocol uses a single message type, wherein a controller broadcasts its ID (assumed to be unique within the communication neighborhood), and its expected load for each 15-minute timeslot in the next 24 hours. In terms of bandwidth, when encoding the controller ID with 4 bytes and the expected load at each timeslot with 2 bytes, the size $L$ of each message amounts to $4 + 2 \cdot 24 \cdot 4$ bytes. Additional payload might result from the characteristics of the technologies used to implement reliable local data communication in the network. On average, each controller broadcasts a message twice per hour (i.e., after each rescheduling). Then, the total expected payload generated by a single controller amounts to $L \approx 10$ Kbytes per day, i.e. roughly 1 bit per second (bps). If $n$ nodes are present in a fully-connected neighborhood, the total amount of generated payload equals to $n \cdot L$.

Up to relatively large values of $n$ this amount of traffic can be reasonably handled by available powerline communication (PLC) or wireless technologies. For example, the set of narrow-band PLC standards promoted by the PRIME Alliance for metering provides a minimum throughput of 21.4 kbps [20], whereas the G3-PLC specification [21], backed by several companies providing inexpensive, interoperable implementations, yields a 33.4 kbps data rate; similar performance can be expected from the P1901.2 standard being developed by the IEEE [22]. When used as a PLC communication infrastructure, any of these technologies would provide enough bandwidth for our system to handle huge communication neighborhoods including thousands of households, which are much larger than necessary. Note that communication requirements only depend on the amount of households in each communication neighborhood, and are not affected by the total amount of neighborhoods. Therefore, the system is inherently scalable to huge deployments, provided that the size of each neighborhood does not exceed the specified limits.

Wireless communication channels may be adopted if a PLC communication infrastructure is not desirable or available. Depending on the specific environment, low-power Wireless Personal Area Network (WPAN) systems such as those implementing the ZigBee specifications may be used to create small communication neighborhoods encompassing a limited amount of households; DSM controllers in different households within the same apartment building [23], [24] or within an area of single-family houses [25] may communicate with each other via these technologies. For example, the Indriya [26] testbed demonstrates wireless links among 127 low-power nodes scattered in a 3-floor university building. In other situations, better connectivity may be provided by using Wireless Local Area Network (WLAN) systems (i.e. the family of 802.11 Wi-Fi standards); for example, TFA [27] is a large scale urban deployment interconnecting 4000 users in a 4km² area. In general, when using wireless links, the physical distance among controllers plays an important role and affects the connectivity topology. The group of considered households is not partitioned in well-defined communication neighborhoods isolated from each other; instead, controllers are connected in a mesh topology, and the propagation of messages depends on the implemented forwarding mechanisms. The proposed coordination approach is well suited for this sort of topologies: however, the experimental evaluation in Section III focuses on PLC-like topologies with well-defined communication neighborhoods.

G. Computational requirements and deployment

The activity of DSM controllers can be decomposed in three main tasks: (a) handling communication with neighboring controllers and aggregating forecast load profiles; (b) building optimization subproblems and executing the resulting schedules; (c) solving the optimization subproblems. Tasks (a) and (b) are computationally straightforward and can be easily handled even by low-end microcontrollers. Point (c) is comparatively harder. In terms of scalability, only task (a) depends on the number $n$ of households in the neighborhood. In fact, once an aggregate load profile is computed, building and solving optimization subproblems is independent on the size of the neighborhood. In our current implementation, a single optimization run requires less than 0.1 seconds of CPU time on a standard laptop. Solving the same problems on embedded hardware will require longer computation times: however, because consecutive optimization runs are triggered at an average interval of 30 minutes, each optimization run may take up to few minutes to complete without detrimental effects on the system.

In the context of the S2G [28] project, field tests are being implemented. The proposed algorithm is deployed to control different types of appliances in 20 households. Initial phases focus on three appliance classes: heat pumps for ambient and water heating; Home Charge Devices for electric vehicle charging [29]; bidirectional batteries, which can be directly controlled through a serial input. In contrast, implementation to common household nonpreemptible appliances, such as dishwashers or washing machines, requires a larger amount of user interaction and is therefore delayed to a later phase, also owing to the limited contribution of such devices to the total household loads.
In an initial test deployment, appliances are directly controlled by a custom-built Home Appliance Controller (HAC) device [28]. Initial tests involve an external computer running the optimization algorithm, and interacting with the HAC (and, in turn, with appliances) by means of the open-standard domotic interoperability framework DomoML [30]. Later, implementation of the optimization algorithm is planned directly inside the HAC device, which is based on the ST Microelectronics Value Line STM32F100 microcontroller platform [31]. Such platform, priced at under 10 US$, is equipped with an ARM Cortex M3 24 MHz CPU, 128 KB of flash memory and 8 KB RAM; the HAC is also equipped with mechanisms for logging and measuring frequency and voltage drop, which may be used as further control inputs, alternative or additional to communicated data. In a longer perspective, algorithms are planned to be integrated directly within the control mechanisms of appliances, to provide finer control and access various monitoring data (such as temperature readings within a water heater, or duration of a dishwasher’s program).

III. EXPERIMENTAL RESULTS

Simulation experiments have been performed on a set 120 households. All households are downstream a single MV-LV transformer. Each experiment is run for a total simulation period $T_{sim}$ equal to 10 days.

For each experiment, we report two measures of performance, which should be minimized: the average energy cost payed considering all households, and an indicator for the fluctuations of the aggregate network load. The former is denoted as $Z_{cost}$ and is easily computed considering a bi-level pricing profile, such as those implemented in Switzerland and Italy [8]. The latter is computed as $Z_{load} = \sum_{t \in T_{sim}} (P_{aggregate}^t \cdot \Delta t)^2$, where $t$ represents a 10-second simulation timestep and $P_{aggregate}^t$ is the aggregate load in $t$. For representing load stability, alternative measures could have been used, such as the amplitude of the large load peaks, the maximum/mean/median loads at predefined peak hours [32], or statistics on voltage drops at different parts of the distribution network. All such measures are in fact correlated, and are minimized when the aggregate load is flattened.

As previously remarked, controllers do not affect the total amount of energy required by the appliances in $T_{sim}$. For a given total amount of energy used in the time period $T_{sim}$, $Z_{cost}$ is minimized if such energy is all used in the low-tariff period (i.e., during the night), whereas $Z_{stability}$ is minimized if the energy is used with a flat load profile. However, because of appliance operating constraints, as well as non-dispatchable loads, none of these optimal load profiles is ever achieved in our simulations.

We considered and simulated a number of different scenarios in which communication is exploited for information sharing. In these cases, a lexicographic optimization policy with energy price is set to as primary objective, with load stability as secondary objective. We divided 120 households in 120/n communication neighborhoods of n households each, and we run simulations for $n = 2, 3, 4, 5, 6, 8, 10, 20, 60, 120$. In the following, these scenarios are indicated with C-n. The C-120 cases implies full communication between all simulated households). We also simulate the scenario Stability-120, where a single-objective optimization of load stability ($z_s$) is implemented with full communication. Results for this scenario are considered as a lower bound to load instability for comparing with previous approaches.

Numeric results are reported after normalization. In particular, $Z_{cost}$ values are linearly rescaled in such a way that $Z_{cost} = 100$ in the Baseline scenario. $Z_{load}$ values are normalized between 0 (for the Stability-120 scenario) and 100 (for the Baseline case).

We report on two sets of experiments: a) DSM applied to electric vehicle (EV) chargers, where we study the effect of DSM algorithms on the distribution of EV charging loads, ignoring other loads; and b) DSM applied to a full set of household appliance loads (in addition to EV chargers).

A. DSM of electric vehicle charging loads

For each of the 120 households, we simulate a single electric vehicle and a resident which makes a reasonable use of it.

---

### Table I

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Normalized $Z_{cost}$</th>
<th>Normalized $Z_{load}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>100</td>
<td>100.0</td>
</tr>
<tr>
<td>Baseline + Randomization</td>
<td>76</td>
<td>18.0</td>
</tr>
<tr>
<td>Cost</td>
<td>62</td>
<td>347.0</td>
</tr>
<tr>
<td>Cost + Randomization</td>
<td>62</td>
<td>81.5</td>
</tr>
<tr>
<td>C-2</td>
<td>62</td>
<td>79.3</td>
</tr>
<tr>
<td>C-3</td>
<td>62</td>
<td>76.6</td>
</tr>
<tr>
<td>C-4</td>
<td>62</td>
<td>75.6</td>
</tr>
<tr>
<td>C-5</td>
<td>62</td>
<td>74.7</td>
</tr>
<tr>
<td>C-6</td>
<td>62</td>
<td>74.5</td>
</tr>
<tr>
<td>C-10</td>
<td>62</td>
<td>74.2</td>
</tr>
<tr>
<td>C-20</td>
<td>62</td>
<td>74.1</td>
</tr>
<tr>
<td>C-120 (full connectivity)</td>
<td>62</td>
<td>73.9</td>
</tr>
<tr>
<td>Stability (full connectivity)</td>
<td>82</td>
<td>0.0</td>
</tr>
</tbody>
</table>

---

**Baseline-random.** Appliances are managed by a controller that randomizes their behavior, still meeting their operational constraints. For example, EVs are charged using a random schedule which at the same time enforces that the vehicle is charged at the due date.

**Cost.** Controllers only implement the single-objective optimization of energy costs ($z_c$).

**Cost-random.** Controllers implement single-objective optimization of energy cost ($z_c$): remaining degrees of freedom in load scheduling – i.e. load shifts which would keep the same value for $z_c$ – are explicitly randomized, still meeting operational constraints.
In particular, we modeled EV usage by considering Gaussian-distributed leaving and arrival times of EVs at the households, with average values at 7:30AM and 7:00PM respectively, and standard deviations of 2.5 and 4 hours. After use, the battery of each EV is depleted by a random amount, uniformly distributed between 10% and 90% of the full charge, and needs to be recharged to full power before the next use of the vehicle, which is set as a due date for the charging job.

A single realization of random variables defining EVs’ use patterns is repeated for all experiments, and a fixed energy state for all appliances is enforced at the beginning and at the end of each experiment. This ensures that the total energy drawn is the same for each experiment, and that different experiments can be meaningfully compared.

Results reported in Table I and Figure 1 show that communication improves network stability even when small neighborhood sizes are considered. Moreover, results show that scenarios ignoring cost optimization (Baseline-random and Stability-120) greatly improve network stability (but achieve reduced cost savings). In fact, these scenarios allow controllers to schedule EV charging at any time of the day; on the contrary, other scenarios – in which cost optimization is enforced – impose to charge vehicles during the low-tariff periods as much as allowed by user constraints, which inevitably creates higher average loads during such periods. Note that in Table I, energy cost is reported in normalized units, with 100 representing the cost payed in the baseline case (i.e., when no controllers are deployed). $Z_{load}$ is also normalized between 0 (scenario Stability-120) and 100 (Baseline).

**Behavior under dynamic pricing profiles**

In a separate experiment, we computed the $Z_{load}$ performance measure in scenario C-4, considering the very dynamic Swedish energy pricing profile (which changes every 15 minutes). This resulted in a normalized $Z_{load}$ value of 345, much larger than $Z_{load} = 75.6$, which was obtained with the swiss bi-level pricing profile. In particular, the same poor performance for network stability was observed regardless of the size of the communication neighborhood, including the full communication case.

This confirms the findings already reported in [8]: under dynamic energy pricing profiles, lexicographic optimization of the primary objective (cost) leaves very little room for improvement of the secondary objective (load stability). On the contrary, cost optimization promotes the emergence of load peaks at times when the price is minimum, which cannot be reduced unless cost objective $z_c$ is allowed to deviate from the optimum. Therefore, particular attention should be put in the design of dynamic demand-response policies.

**B. DSM of household appliance loads**

In a different set of experiments, we investigated the effects of DSM algorithms acting, other than on EVs, also on some household loads. In this case, the performance is evaluated on the aggregate load of the residential neighborhood, which includes all residential energy use.

For each household we simulate an EV with the same characteristics as described before. In addition, the DSM algorithm can control up to two energy buffer appliances (one water heater and, in some households, an air conditioning system), as well as one or two non-preemptible appliances.
TABLE II
ENERGY COST AND LOAD INSTABILITY FOR HOUSEHOLD APPLIANCE AND EV LOADING TASKS.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Normalized $Z_{\text{cost}}$</th>
<th>Normalized $Z_{\text{load}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>100</td>
<td>100.0</td>
</tr>
<tr>
<td>Baseline + Randomization</td>
<td>99</td>
<td>64.1</td>
</tr>
<tr>
<td>Cost</td>
<td>88</td>
<td>33.4</td>
</tr>
<tr>
<td>Cost + Randomization</td>
<td>88</td>
<td>22.8</td>
</tr>
<tr>
<td>C-2</td>
<td>88</td>
<td>8.9</td>
</tr>
<tr>
<td>C-3</td>
<td>88</td>
<td>4.4</td>
</tr>
<tr>
<td>C-4</td>
<td>88</td>
<td>4.4</td>
</tr>
<tr>
<td>C-5</td>
<td>88</td>
<td>3.9</td>
</tr>
<tr>
<td>C-6</td>
<td>88</td>
<td>4.1</td>
</tr>
<tr>
<td>C-10</td>
<td>88</td>
<td>3.8</td>
</tr>
<tr>
<td>C-20</td>
<td>88</td>
<td>3.6</td>
</tr>
<tr>
<td>C-120 (full connectivity)</td>
<td>88</td>
<td>3.4</td>
</tr>
<tr>
<td>Stability (full connectivity)</td>
<td>93</td>
<td>0.0</td>
</tr>
</tbody>
</table>

(washing machine or dishwasher). The remaining appliances (fridges, cooking appliances, lighting, entertainment) are considered non-controllable, and DSM algorithms cannot affect their energy use. However, DSM algorithms can measure the total power load of the household and therefore learn an expected load curve of non-controllable appliances; such learned expected load is included in the forecast of future loads which is used during optimization (and communicated to neighbors).

The same scenarios considered before are also simulated in this setting. The results are reported in Figure 2 and Table II. Even very small communication neighborhoods (C-2) significantly improve load stability with respect to scenarios without communication. However, further increasing the neighborhood size has very limited benefits. The Stability-120 scenario further improves overall network stability at the expense of lower cost savings.

In the Baseline case, the aggregated load profile results in the characteristic load profile with peaks in the late morning and late afternoon, as visible in Figure 3. The relatively large loads during the night are due to EV charging.

When the algorithms optimize energy cost without communicating with each other, one observes very clear peaks at times when the bi-level tariff changes: this is the effect of the DSM algorithms collectively filling energy buffers as much as possible before cost increase. Load randomization (subject to cost optimization) mitigates this problem, as shown in Figure 2.

Communication, even with very small neighborhoods, significantly improves all network stability measures: the resulting load profile in Figure 3 shows that most load peaks have disappeared due to coordination among nearby households. Neighborhoods of 3-4 households appear to be sufficient to achieve near-optimal network stability performance, while increasing neighborhoods beyond this size leads to negligible improvements. Figure 3 also shows how effectively the load profile can be flattened by instructing DSM algorithms to ignore energy costs. Effective coordination of neighboring households is demonstrated in Figure 4: note how households in the same neighborhood tend to alternate high-power loads (bright areas) in such a way that they rarely draw large amounts of power at the same time.

In our system, interaction is limited to controllers within the same communication neighborhood. Therefore, the results presented above can generalize to LV networks of different sizes: the only significant difference is that a larger network will contain more communication neighborhoods, and a smaller one will contain fewer. The stabilizing effect of the proposed system within each communication neighborhood will be mostly unaffected. This is verified by means of the experiment reported in Figure 5. The figure shows the normalized load instability for a subset of the scenarios considered in Figure 2; however, simulations are repeated considering each of three differently-sized LV networks (60 households, 120 households, 480 households). The stabilizing performance of the system is not significantly affected by the LV network size, but mainly depends on the communication ability and the size of the communication neighborhood.

Depending on the adopted communication technique, some of the messages exchanged within neighborhoods may be lost or may only reach a subset of the recipients. Consider the case in which a message from a given controller $C$ is lost: the only consequence is that neighboring controllers will...
optimize their own loads ignoring the latest load forecast of controller $C$ (but still considering forecasts from other controllers, and, if known, the forecast of $C$ that was broadcast during the previous optimization round). In practice, because optimization is repeated on average once every 30 minutes, losing a single message leads to negligible losses. In scenario $C-2$ of Table II, the normalized load instability measure grows from 8.8 to 9.5 when the packet loss probability is increased from 0 to 10%, and reaches 14.2 if the packet loss probability is as high as 50%. Such performance is still significantly better than the performance obtained without communication (22.8). Similar measures are observed with larger communication neighborhoods.

**IV. CONCLUSIONS**

We addressed the problem of decentralized demand-side management of household loads, in a residential area under a single MV-LV transformer. Every household is equipped with a controller implementing a multi-objective optimization algorithm, accounting for energy cost and stability of the aggregate load. The latter objective is optimized in a distributed fashion by means of an asynchronous communication infrastructure which allows coordination among households belonging to the same neighborhood.

We considered neighborhoods of different size and observed that the aggregate load at the transformer is significantly improved – with smaller peaks and a more even distribution of load – even when neighborhoods as small as 2 or 3 households are considered. This implies that even low cost technologies, i.e., communication technologies with low bitrate and communication range, are able to obtain very good performances with respect to technologies allowing for broader communication (all-to-all).

Currently, field tests are being setup involving 20 households in the Swiss town of Mendrisio, in the context of the Swiss2Grid project [28].
ACKNOWLEDGMENTS

This work has been sponsored by Swiss Electric Research and the Swiss Federal Office of Energy (BFE project nr.103162).

REFERENCES


Alessandro Giusti received the Master’s degree in Computer Science in 2005 and a Ph.D. in Computer Science in 2009 from Politecnico di Milano, Italy. He is currently a researcher at the Dalle Molle Institute for Artificial Intelligence (IDSIA), a research institute of the University of Applied Sciences of Southern Switzerland (SUPSI) and of the University of Lugano (USI). His research is focused on distributed algorithms, robotics, biological image processing and visualization.

Matteo Salani received the Master’s degree in Computer Science in 2001 and a Ph.D. in Computer Science in 2006 from Università degli Studi di Milano, Italy. He was an assistant professor (PostDoc) at Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland. Currently, he is a researcher at IDSIA. His research interests deal with mathematical programming and optimization. In particular, he is interested in exact algorithms for combinatorial problems arising in transportation and logistics.

Gianni Di Caro received a degree in Physics from the University of Bologna (Italy), and a Ph.D. in Applied Sciences from the Université Libre de Bruxelles, Belgium. Currently he is senior researcher at IDSIA. He has co-authored more than 100 peer-reviewed works focusing on networking, optimization, bio-inspired algorithms, and distributed robotics.
Andrea E. Rizzoli received the Master’s degree in Control Engineering in 1989 and a Ph.D. in Control Engineering and Informatics in 1993 from Politecnico di Milano, Italy. He was a PostDoc researcher at CSIRO, Australia. Presently, he is a senior research scientist at IDSIA and a professor at SUPSI. His current research interests focus on methodologies and techniques for model re-use and encapsulation in different domains, from environmental systems to transportation and logistics systems. He is a founding member and the incumbent President of the International Environmental Modelling and Software Society, whose aim is to develop and use environmental modelling and software tools to advance the science and improve decision making with respect to resource and environmental issues.

Luca M. Gambardella is director of IDSIA since 1995. His major research interests and publications are in the area of Ant Colony Optimization meta-heuristic for combinatorial optimization problems, simulation, machine learning and swarm robotics. He acquires and manages practical and theoretical projects with companies, the European Commission, and the Swiss CTI and SNF.