A Distributed Constraint Optimization (DCOP) Approach to the Economic Dispatch with Demand Response

Ferdinando Fioretto
University of Michigan
Ann Arbor, MI, USA
fioretto@umich.edu

William Yeoh
New Mexico State University
Las Cruces, NM, USA
wyeoh@cs.nmsu.edu

Enrico Pontelli
New Mexico State University
Las Cruces, NM, USA
epontell@cs.nmsu.edu

Ye Ma
Siemens Industry Inc.
Minnetonka, MN, USA
yema@siemens.com

Satishkumar J. Ranade
New Mexico State University
Las Cruces, NM, USA
sranade@ad.nmsu.edu

ABSTRACT

With the growing complexity of the current power grid, there is an increasing need for intelligent operations coordinating energy supply and demand. A key feature of the *smart grid* vision is that intelligent mechanisms will coordinate the production, transmission, and consumption of energy in a distributed and reliable way. *Economic Dispatch* (ED) and *Demand Response* (DR) are two key problems that need to be solved to achieve this vision. In traditional operations, ED and DR are implemented separately, despite the strong inter-dependencies between these two problems. Therefore, we propose an integrated approach to solve the ED and DR problems that simultaneously maximizes the benefits of customers and minimizes the generation costs, and introduce an effective multi-agent-based algorithm, based on *Distributed Constraint Optimization Problems* (DCOPs), acting on direct control of both generators and dispatchable loads. To cope with the high complexity of the problem, our solution employs *General Purpose Graphical Processing Units* (GPGPUs) to speed up the computational runtime. We empirically evaluate the proposed algorithms on standard IEEE bus systems and test the stability of the proposed solution with a state-of-the-art power system simulator on the IEEE 30-bus system.

Keywords

DCOP; Smart Grid; GPGPU

1. INTRODUCTION

The current electricity grid is gradually transforming, focusing on harnessing the potential management from the demand side with the support of widespread decentralized energy resources and active customer participation [8]. To support this transformation, within the *smart grid* vision, intelligent agents will coordinate the production, transmission, and consumption of energy in a distributed and reliable way.

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*
Copyright © 2017, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

Two problems that need to be solved to achieve this vision are economic dispatch and demand response. *Economic dispatch* (ED) [4] optimizes generation schedules to meet predicted load conditions with the lowest cost possible. This is inherently a difficult problem due to (1) the inter-dependencies between the various generator settings; (2) the need to coordinate generators across multiple power plants; and (3) the need to respond promptly to changes in the load. Typically, generation schedules are updated in intervals of 10 minutes or more [33]. Due to prediction errors, communication delays, and changes of operating conditions, real-time adjustments of power generation are necessary, and are typically conducted by automatic generation control (AGC) and local frequency control [15]. Additionally, under high penetration of renewable generation, the forecast pattern of the power production can be highly inaccurate so that adopting solely conventional AGC systems is neither feasible to ensure stable operating conditions nor economically efficient [38].

In response to this challenge, *Demand Response* (DR) [22] services can be used to ensure stable operating conditions (i.e., *Reliability DR Resource*), as well as reducing the cost of ancillary services. DR resources can be planned to be dispatched at different time scales, from supporting frequency regulation—requiring DR loads to respond in the order of few seconds—to correcting phase imbalance on the feeders—which requires responses in intervals of several minutes—and supporting cold load operations by restraining load demand—every several minutes to an hour intervals [14]. Some utilities (e.g., Midwest ISO) have built the infrastructure to enable some of these approaches [32].

When these adjustments are enforced, the energy efficiency of the system may deteriorate, as the process does not take DR applications into account. Indeed, in current system operations, ED and DR are typically solved in isolation, despite the clear inter-dependencies between them—the output of DR (e.g., resulting aggregated load) is the input of ED (e.g., amount of power to be produced by the generators), and the constraints of ED (e.g., cost of activating a generator) translate to some of the objectives in DR (e.g., cost per kWh consumed) [8]. To address this issue, researchers[1] recently started investigating solutions to bridge the gap between the different time-scales at which ED and DR are processed and by seamlessly integrating the two op-

[1]Including industry (e.g., http://www.innovari.com/).

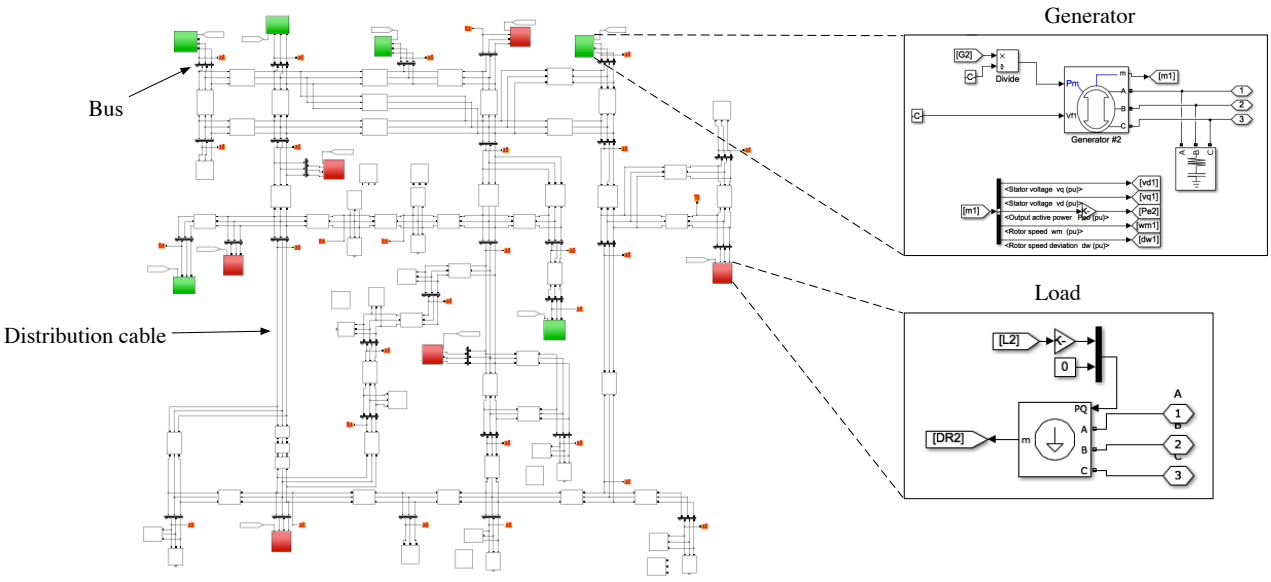

Figure 1: A Power Network Example (IEEE 30 Bus System) (left), and an electromechanical generator and dispatchable load as implemented in the Matlab *SimPowerSystems* simulator (right).

erations together [18]. Additionally, under high penetration of distributed energy resources and increasing customer participation, a challenge for ED and DR is the management of large amounts of scattered data [21]. Most existing solutions for ED and DR are centralized and they require processing large amounts of private data.

Thus, in this paper, we study an integrated and distributed approach for the ED and the DR problems, with the goal of benefiting both energy producers and consumers. Our contribution is fourfold: **(1)** We introduce a power network model that integrates ED and DR and simultaneously maximizes the benefits of customers while minimizing the generation costs; **(2)** We propose a multi-agent based approach that is based on the *Distributed Constraint Optimization Problem* (DCOP) formulation; **(3)** We employ *General Purpose Graphical Processing Units* (GPGPUs) to cope with the high complexity of the problem and speed up our algorithms; and **(4)** We empirically evaluate our algorithms on several standard IEEE bus systems and demonstrate the effectiveness of the returned solutions in terms of system stability on a state-of-the-art power system simulator.

The rest of the paper is organized as follows. In the next section, we introduce an optimization model, called *D-EDDR*, which integrates both the ED and the DR problems. Section 3 reviews DCOPs and Dynamic DCOPs, and maps the proposed D-EDDR problem to a dynamic DCOP. An algorithm to solve such problem is then presented in Section 4. Due to scalability issues, we propose a relaxation of the D-EDDR problem in Section 5 and describe an algorithm to solve the proposed relaxed problem in Section 6. Additionally, we describe how this algorithm makes use of GPGPUs to speed up the agents' resolution. Section 7 discusses the theoretical properties of the algorithms presented. We present related work in Section 8 and summarize our evaluation of the proposed algorithms in Section 9. Finally, Section 10 concludes the paper.

2. DISTRIBUTED ECONOMIC DISPATCH WITH DEMAND RESPONSE (D-EDDR)

We now introduce the *Distributed Economic Dispatch with Demand Response* (D-EDDR) problem, which integrates both ED and DR problems. We describe related ED and DR solution approaches in Section 8.

A power grid can be viewed as a network of nodes (called *buses* in the power systems literature) that are connected by distribution cables and whose loads are served by (electromechanical) power generators. Typically, a group of such power generators are located in a single power station, and a number of power stations are distributed in different geographic areas. Such a power grid can be visualized as an undirected graph $(\mathcal{V}, \mathcal{E})$, in which buses (in $\mathcal{V}$) are modeled as graph nodes and transmission lines (in $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$) as edges. This representation captures the ability of the current to flow in both directions in a circuit. Figure 1 illustrates a representation of the IEEE-30 Bus System, highlighting the components described above.

We consider an n-bus power system, and we assume that each bus injects and withdraws power through a generator $g \in \mathcal{G}$ and a load $l \in \mathcal{L}$, respectively, where $\mathcal{G}$ and $\mathcal{L}$ are, respectively, the set of generators and loads in the problem. Load buses can be *dispatchable* or *non-dispatchable*, based on whether it is possible to defer a portion of the load. In Figure 1, the generators are colored green, while the dispatchable loads are colored red.

We model each bus as an agent, capable of making autonomous decision on its power consumption and generation. W.l.o.g., we assume that there is *exactly* one generator and one load in each bus. The case with multiple loads and generators per bus can be easily transformed into this simpler model by precomputing the best operational conditions for each output combination of loads and generation power.

When a load difference is revealed to the power system, the ED problem is to regulate the output of the appropriate units so that the new generation output meets the new load

and the generators are at economic dispatch (i.e., running efficiently according to some objective function). Near real time power consumption monitoring from smart meters allows short-term load prediction, which can supply the smart grid with predictions on power consumption levels.

The D-EDDR problem with a time horizon H is formulated as follows:

$$\textbf{maximize} \sum_{t=1}^{H} \alpha^t \left(\sum_{l \in \mathcal{L}} u_l(L_l^t) - \sum_{g \in \mathcal{G}} c_g(G_g^t) \right) \quad (1)$$

where $\alpha^t \in (0, 1]$ is a discount parameter that captures the uncertainty in future predictions. Specifically, it is the probability associated with load prediction at time t, determined by the forecaster. L_l^t and G_g^t represent, respectively, the decisions upon the amount of power withdrawn by load l and the amount of power injected by generator g, at time t. The above objective includes the cost functions of the generators as well as the utility functions of the customers' loads, which are defined, respectively, as:

$$c_g(G_g^t) = \alpha_g G_g^t + \beta_g (G_g^t)^2 + |\epsilon_g \sin(\phi_g (G_g^{min} - G_g^t))| \quad (2)$$

$$u_l(L_l^t) = \begin{cases} \beta_l L_l^t - \frac{1}{2}\alpha_l (L_l^t)^2 & \text{if } L_l^t \leq \frac{\beta_l}{\alpha_l} \\ \frac{1}{2}\frac{(L_l^t)^2}{\beta_l} & \text{otherwise} \end{cases} \quad (3)$$

as defined in [29] and [2], respectively, with $\alpha_g, \beta_g, \epsilon_g, \phi_g, \alpha_l, \beta_l$ being coefficients in $\mathbb{R}$. The presence of *valve-point* effects[2] for the generators are captured by considering a sinusoidal term in the quadratic cost function [34], which makes the optimization problem non-convex.

The problem is subject to the following constraints:

$$G_g^{min} \leq G_g^t \leq G_g^{max} \qquad \forall g \in \mathcal{G}; 0 < t \leq H \quad (4)$$

$$L_l^{min} \leq L_l^t \leq L_l^{max} \qquad \forall l \in \mathcal{L}; 0 < t \leq H \quad (5)$$

$$L_l^t \leq \dot{L}_l^t \qquad \forall l \in \mathcal{L}; 0 < t \leq H \quad (6)$$

$$\sum_{l \in \mathcal{L}} L_l^t - \sum_{g \in \mathcal{G}} G_g^t = 0 \qquad 0 < t \leq H \quad (7)$$

$$\vec{L}_{\mathcal{L}}^t - \vec{G}_{\mathcal{G}}^t - \hat{B}\vec{\theta}^t = 0 \qquad 0 < t \leq H \quad (8)$$

$$|f_{ij}^t| \leq f_{ij}^{max} \qquad \forall (i,j) \in \mathcal{E}; 0 < t \leq H \quad (9)$$

$$f_{ij}^t = B_{ij}(\theta_i^t - \theta_j^t) \qquad \forall (i,j) \in \mathcal{E}; 0 < t \leq H \quad (10)$$

$$G_g^{t+1} = G_g^t + \Delta_{p_g}^t \qquad \forall g \in \mathcal{G}; 0 < t \leq H - 1 \quad (11)$$

$$-\Delta_{p_g}^{max} \leq \Delta_{p_g}^t \leq \Delta_{p_g}^{max} \qquad \forall g \in \mathcal{G}; 0 < t \leq H \quad (12)$$

where $\dot{L}_l^t$ is the predicted maximum load at time step t. Eqs. (4) and (5) express the lower and upper bounds for power generation and load consumption at each generation and load unit. These bound are derived by the physical characteristics of generators and loads. Eq. (6) restricts the loads to be within the predicted load limits. Eq. (7) expresses the power supply-demand balance: The amount of power generated must match the amount of power requested at each time. Eq. (8) is a system of linear constraints describing the *DC power flow*. These constraints describe the power flow through each transmission line which depend on the power injections (from the generators), the power withdrawals (from the loads), and the physical characteristics of the transmission lines, which are captured by the entries in the matrix $\hat{B}$. $\hat{B}$ is an n-dimensional matrix whose entries B_{ij} describe the *admittance* associated to the transmission line connecting buses i and j.[3] In power systems, the *admittance* is a measure of how easily a circuit will allow a current to flow. The DC model is an extensively studied approximation to a full (nonlinear) AC power flow [37]. The DC load flow relates real power to voltage phase angle, ignores reactive power issues, and assumes voltages are close to normal. $\vec{G}^t$, $\vec{L}^t$, and $\vec{\theta}^t$ are, respectively, the n dimensional vectors of power generated, the power injected, and the bus phase angles, associated to each unit in $\mathcal{V}$ at time t. The phase angles describe the phase difference between the voltage and current at the bus.

Eqs. (9) and (10) restrain the network flow over the line connecting buses i and j, with f_{ij}^{max} denoting the line flow limit. Such constraints relate the amount of power being carried over each transmission line to the net injections and withdraws of each bus (in turn, related to the bus voltage phase angles). Eqs. (11) and (12) express ramp constraints for each generator. That is, the maximum incremental power that can be supplied or reduced in one time step, and depends on the mechanical characteristics of the generator. For more details on electrical power systems and power flow study we refer the interested reader to [37].

To ensure stable operation of generators, *prohibited operating zones* (POZs) are used to forbid the generators to produce power within certain intervals [2, 29]. POZs are formalized with the following constraint (where $g_1, \ldots, g_{z_g}$, are the POZs for generator g):

$$\begin{cases} G_g^{min} \leq G_g^t \leq G_{g_1}^{min} \\ G_{g_{i-1}}^{max} \leq G_g^t \leq G_{g_i}^{min}, \quad \text{if } i = 2, \ldots, z_g \\ G_{g_{z_g}}^{max} \leq G_g^t \leq G_g^{max} \end{cases} \quad (13)$$

Notice that the D-EDDR problem becomes non-smooth due to the presence of POZs; coupled with the non-convex costs (Eq. (2)) in the objective function, it makes the D-EDDR optimization problem a particularly challenging one.

To empirically verify our hypothesis that an integrated ED with DR approach will yield better results than the traditional approach of only considering ED, we ran a preliminary experiment on an IEEE 30-bus system, where we optimally solve the *convex* optimization problem of D-EDDR, and compare it against the optimal solution when DR is not considered. Figure 2 shows the effect of applying the integrated solution approach, yielding a larger utility.

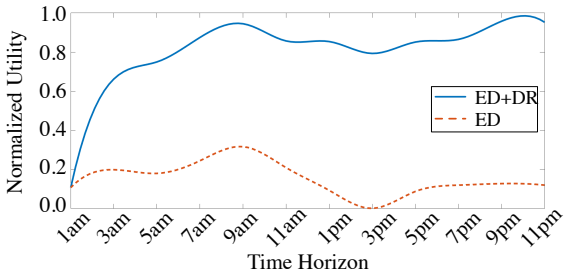

Figure 2: Utility of Isolated ED vs Integrated ED with DR

[2] When a sequence of generator steam valves are opened, there is a rippling effect on the generator's power-cost curve, called *valve-point* effect.

[3] $B_{ij} = 0$ if no transmission line exists between buses i, j.

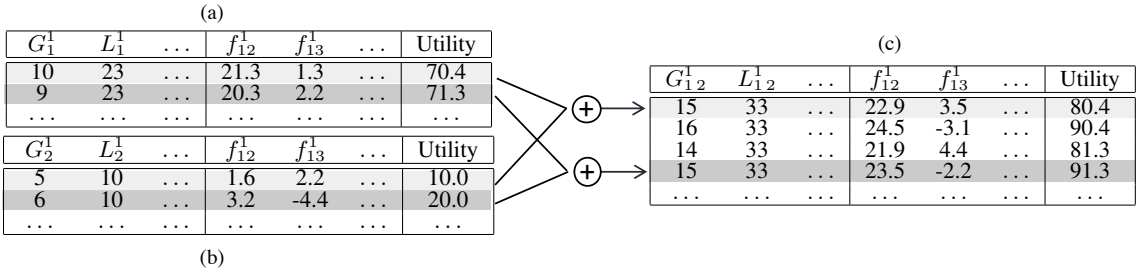

Figure 3: Example Utility Tables.

3. DYNAMIC DCOP REPRESENTATION

A *Distributed Constraint Optimization Problem* (DCOP) [20, 23] is defined by $\langle \mathcal{X}, \mathcal{D}, \mathcal{F}, \mathcal{A}, \alpha \rangle$, where: $\mathcal{X} = \{x_i\}_{i=1}^n$ is a set of *variables*; $\mathcal{D} = \{D_i\}_{i=1}^n$ is a set of finite *domains*, where D_i is the domain of variable x_i; $\mathcal{F} = \{f_i\}_{i=1}^m$ is a set of *utility functions* (or *constraints*), where each k-ary utility function $f_i : D_{i_1} \times \ldots \times D_{i_k} \mapsto \mathbb{R} \cup \{-\infty\}$ specifies the utility of each combination of values of variables in its *scope* (i.e., $x_{i_1}, \ldots, x_{i_k}$); $\mathcal{A} = \{a_i\}_{i=1}^p$ is a set of *agents* and $\alpha : \mathcal{X} \rightarrow \mathcal{A}$ maps each variable to one agent. A *solution* is a value assignment for all the problem variables satisfying the problem constraints. Its utility is the sum of the evaluations of all utility functions on it. The goal is to find a utility-maximal solution. A *Dynamic DCOP* [24] is a sequence of DCOPs $\langle P_1, P_2, \ldots \rangle$, where each P_t is associated with a particular time step t. The goal is to find a utility-maximal solution for each DCOP in the sequence.

We now describe how one can model the D-EDDR problem as a Dynamic DCOP. The time steps in the Dynamic DCOP correspond to the time steps of the D-EDDR problem. Thus, both the D-EDDR problem and the corresponding Dynamic DCOP have a horizon of H. Each DCOP P_t in the Dynamic DCOP has exactly the same agents, variables, and domains, where the agents $\mathcal{A}$ correspond to the network buses; each agent $a \in \mathcal{A}$ controls two variable: L_l^t ($l \in \mathcal{L}$) and G_g^t ($g \in \mathcal{G}$)—i.e., a generator and a load; and the domain of each variable $D_i \in \mathcal{D}$ corresponds to the possible power that can be injected (Eq. (4)), if it is a generator, or withdrawn (Eq. (5)), if it is a load. Additionally, the utility functions include hard constraints that represent Eqs. (6), (7), (12), and (13), which are constraints within each DCOP P_t, and hard constraints that represent Eq. (11), which are constraints between DCOPs P_t and P_{t+1} of subsequent time steps. Finally, the objective function in Eq. (1) is decomposed into $|\mathcal{L} \cup \mathcal{G}|$ unary constraints, one for each agent's variable—each agents controlling a load $l \in \mathcal{L}$ and a generator $g \in \mathcal{G}$ have constraints $\alpha^t \cdot u_l(L_l^t)$ and $-\alpha^t \cdot c_g(G_g^t)$.

This current formulation is incomplete as it does not include the constraints on the bus voltage phase angles $\vec{\theta}^t$ and the line flows f_{ij}^t in Eqs. (8)–(10). Both the phase angles and the line flows are continuous variables in the optimization problem. However, since they are not decision variables (like the load and generator variables), we formulate them as *environment* variables, whose values are a function of the decision variables in the problem. The constraints on these variables can thus be checked for consistency given a complete assignment for the decision variables. Note that the determination of the values of the environment variables and whether their constraints are violated cannot be enforced by any unassisted single agent—as they depend on the values of *all* the decision variables. Thus, one can view these environment variables and their constraints as a single complex global constraint in each DCOP of the Dynamic DCOP.

4. DEEDS

We now introduce the *Distributed Efficient ED with DR Solver* (DEEDS), a dynamic programming based approach that can be used to solve D-EDDR problems. DEEDS bears similarities with DPOP [23], which is used to solve DCOPs. We first describe a naive version of the algorithm that solves the problem exactly. However, due to high complexity of the EDDR problem, this approach is unsuitable to solve even the smallest instance of our problems, even if we consider a single time step horizon. We then introduce an approximated version which offers better scalability.

DEEDS is composed of the following four phases:

[Phase 1] Pseudo-tree Generation: DEEDS constructs a pseudo-tree using any existing a distributed pseudo-tree construction algorithms, such as Distributed DFS [13].

[Phase 2] Utility Initialization: Each agent a constructs its utility table U_a, where:

- The first $2 \cdot H$ columns of the table are all possible combinations of the generation and loads values of that agent at the different time steps (i.e., G_a^1, L_a^1, G_a^2, L_a^2, ...).
- The next $|F| \cdot H$ columns of the table correspond to the contribution of that agent to all the line flows f_{ij}, given its *net injections* (the amount of power generated menus the one consumed at that bus) listed in the first $2 \cdot H$ columns. The generation and loads determine the possible bus voltage phase angles through Eq. (8), which determine the contributions to the line flows through Eq. (10).
- Finally, the last column is the utility associated to that agent, and calculated according to the objective function in Eq. (1).

Figures 3(a) and 3(b) illustrate two example utility tables of two agents, a_1 and a_2. During this phase, the agents also prune the rows of its utility table that violates the constraints in Eqs. (6), (11)–(13).

[Phase 3] Utility Propagation: Once the utility tables are generated, each leaf agent sends its utility table to its parent, which then performs an *aggregation* operation between the tables that it received from all its children and its own table. Such process creates a new aggregated utility table, where each row is the sum of one row from each of the source tables. Thus, these operations differ from that of DPOP. More specifically, assume there are k tables to be aggregated (e.g., the agent is aggregating the tables from

its $k-1$ children and its own table), and let $\vec{x}^i_r$ denote the r-th row of elements in the i-th table. Then, the aggregated table has the following $\sum_i \vec{x}^i_{r_i}$ elements in each of its row:

- The first $2 \cdot H$ columns contain the total power injections and withdrawals of the subtree at the aggregating agent.
- The next $|F| \cdot H$ columns contain the power flow values associated to the corresponding net injection.[4]
- The last column contains the sum of the utilities.

Importantly, the number of columns stays constant in the aggregation step, whilst the number of rows in the resulting table is the number of combinations of rows of the k utility tables. More specifically, aggregating tables for two agents a_1 and a_2, whose initial sizes are $|T_i| = [(G^{max}_{a_i} - G^{min}_{a_i}) \cdot (L^{max}_{a_i} - L^{min}_{a_i})]^H$, for $i = 1$ and 2, results in a new table with $|T_1 \times T_2|$ rows.

Figure 3(c) shows the result of aggregating the utility tables in Figures 3(a) and 3(b), where we assume that agent a_1 is the child of agent a_2.

Each non-root agent then sends the aggregated table up to its parent, which repeats this process. The utilities thus propagate up the pseudo-tree. Once the root agent produces the aggregated utility table, it prunes from it the rows violating the remaining global constraints in Eqs. (7)–(10).

[Phase 4] Value Propagation: The root agent chooses its value corresponding to the optimal row in its aggregated table—the row with the maximum utility—and sends this value and the row ID of the child's utility table used to construct the optimal row to each of its children agents. Each child agent repeats the process and propagates the values down the pseudo-tree.

5. RELAXED D-EDDR

Unfortunately, DEEDS does not scale to problems of interest due to the excessive amount of resources required, prohibiting the use of off-the-shelf complete DCOP algorithms.[5] The problem is with the size of the utility tables generated and aggregated. If, for each combination of generation and load values, there is a unique contribution to the line flows, then the number of rows in the utility table initialized in Phase 2 is exactly the number of such combinations (i.e.,$[(G^{max}_a - G^{min}_a) \cdot (L^{max}_a - L^{min}_a)]^H$) for agent a. There are multiple possible contributions to the line flows per combination of generation of load values. Thus, the number of rows in the utility table cannot be bounded (since the line flows are continuous values). This problem is exacerbated by the aggregation procedure that creates an exponential number of additional rows.

Let us introduce a relaxation of the D-EDDR problem and a relaxed version of DEEDS to address this issue. The reason why all the possible contributions to the line flows for each combination of generation and load values need to be stored is due to the non-monotonicity of the line flow constraint; the agents are unable to verify whether the line flows violate the global constraint until the contributions for all agents are aggregated, which is done once the utility table reaches the root agent. In our example in Figure 3(c) the highlighted rows describe the same injections and withdrawals, however their contributions to the line flows differ, thus both tables rows need to be stored.

To cope with this issue, our D-EDDR relaxation moves the global constraint on the line flows to the objective function, transforming it into a soft constraint with a penalty on the degree of its violation. The resulting objective function is:

$$\sum_{t=1}^{H} \alpha^t \left(\sum_{l \in \mathcal{L}} u_l(L^t_l) - \sum_{g \in \mathcal{G}} c_g(G^t_g) - \sum_{(i,j) \in \mathcal{E}} \lambda^t_{ij} \right) \qquad (14)$$

with λ^t_{ij} as:

$$\lambda^t_{ij} = \begin{cases} \omega \left(|f^t_{ij}| - f^{\max}_{ij} \right)^2 & \text{if } \frac{|f^t_{ij}|}{f^{\max}_{ij}} > m \\ 0 & \text{otherwise.} \end{cases}$$

where $\omega \in \mathbb{R}^+$ and $m \in [0, 1]$ are parameters of the problem. This approach is similar to the one used by researchers in the power systems community [31, 35]. Such relaxation is warranted, as regulations typically allow flows to be outside a prescribed range for a very small amount of time (e.g., few minutes). Note that the solution for the relaxed D-EDDR approaches that of D-EDDR as ω tends to 0.

6. RELAXED DEEDS

We now describe Relaxed DEEDS (R-DEEDS), an iterative version of DEEDS that solves the relaxed D-EDDR problem optimally. Figure 4 shows the flow chart of the algorithm.

Similar to DEEDS, R-DEEDS agents also construct a pseudo-tree (Phase 1) and initialize their utility tables (Phase 2). Since the global constraint on the line flows are now in the objective function, each agent can select the line flow that maximizes its utility for each combination of its generation and load values. Thus, the number of rows in its initialized utility table is *at most* the number of such combinations—$(G^{max}_a - G^{min}_a)^H$, or $(L^{max}_a - L^{min}_a)^H$, for agent a, with $a \in \mathcal{G}$, or $a \in \mathcal{L}$, respectively.

Additionally, during the aggregation step, the number of rows of the table, that is a result of aggregating the tables for two agents a_1 and a_2, is $[((G^{max}_{a_1} + G^{max}_{a_2}) - (G^{min}_{a_1} + G^{min}_{a_2})) \cdot ((L^{max}_{a_1} + L^{max}_{a_2}) - (L^{min}_{a_1} + L^{min}_{a_2}))]^H$, which is thus substantially lower than $|T_1 \times T_2|$, with T_1 and T_2 representing the values for the tables of agents a_1 and a_2, respectively. This observation is shown in our example in Figure 3(c). The row highlighted dark gray will be preferred over the light gray one, as its has the greatest utility between such two rows, and the value combinations in the light gray row can hence be eliminated.

R-DEEDS agents propagate the utility table up to the root agent analogously to agents in DEEDS (Phase 3). However, differently from DEEDS, when the root agent checks for satisfiability of the global constraints in Eqs. (7)–(10),

- If none of the rows satisfy all the constraints, it suggests that the problem is insufficiently relaxed. Thus, it increases ω by a factor of 2 based on the difference with respect to its value in the last iteration,[6] propagates this information to all agents, and informs them to reinitialize their utility tables with the new updated ω, and solve the new relaxed problem in a new iteration.

[4] The net power injections at a bust are linearly related to the DC line power flows, making such operation feasible.

[5] We used DPOP, a state-of-the art dynamic programming-based algorithm.

[6] The larger the value of ω, the more relaxed the problem. The relaxed problem is identical to the original one if $\omega = 0$.

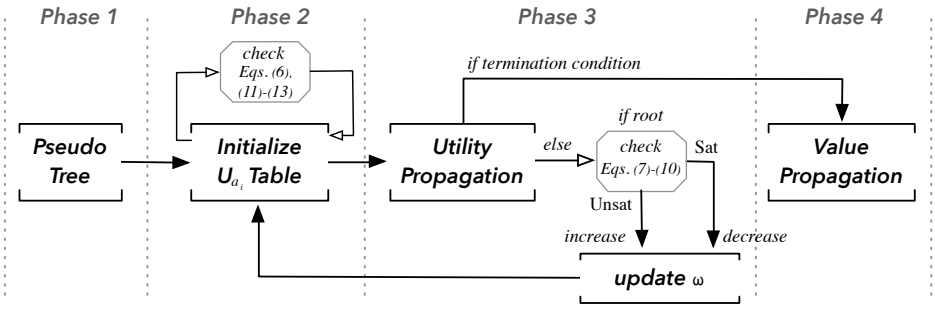

Figure 4: Flow Chart of Relaxed DEEDS

- If there is a row satisfying all the constraints, then the problem may be overly relaxed. Thus, it decreases ω by a factor of 2 based on the difference with respect to its value in the last iteration, and this new relaxed problem is solved in a new iteration.

This process repeats itself until some criteria is met (e.g., a maximum number of iterations, convergence in the solution). Finally, the algorithm runs the value propagation phase in the same way as DEEDS (Phase 4) before terminating.

One can further improve the algorithm's scalability by not considering all time steps together. We propose a further approximation that solves the problem H_{opt} time steps at a time. Thus, it first solves the subproblem with the first H_{opt} time steps. Next, it solves the following subproblem with the next H_{opt} time steps, and so on, until the entire problem is solved. The satisfiability of the constraints between the subsequent H_{opt} subproblems is ensured in Phase 2.

6.1 GPU Optimization

General Purpose Graphics Processing Units (GPGPUs) are multiprocessor devices, offering hundreds of computing cores and a rich memory hierarchy. A parallel computation is described by a collection of procedures executed by several *threads*, which have access to several memory levels, each with different properties in terms of speed and capacity. GPGPUs support *Single-Instruction Multiple-Thread* (SIMT) processing, where the same instruction is executed by different threads, while handling different data.

A number of the operations of the algorithm can be sped up through parallelization using GPGPUs. In particular, the initialization and consistency checks of the the different rows of the utility table can be done independently from each other. Additionally, the aggregation of the different rows of the utility tables can also be computed in parallel, fitting well the SIMT processing paradigm. In our example in Figure 3, all the rows of Table (c) can be computed in parallel. Thus, we adopt a scheme similar to that proposed in [9] to implement a GPGPU version of R-DEEDS.

7. COMPLEXITY ANALYSIS

We report below a theoretical analysis on the network load, messages size, and agent complexity provided by the proposed DEEDS algorithms. The *network load* and *messages size* are defined, respectively, as the total number of messages exchanged by the agents, and as the size of the largest message exchanged by the agents during problem resolution. Since our algorithms rely on an inference-based procedure, the agent's complexity (i.e., the maximal number of operations performed by the agents) is equivalent to the size of the largest message exchanged.

Theorem 1. *The network load of (R)-*DEEDS *is linear in the number of iterations and the number of buses in the problem.*

Proof. At each iteration, the *Utility Initialization* phase is carried by each (R)-DEEDS agent independently, with no need of coordination. When the leaf agents complete this phase, they start the *Utility Propagation* phase, which requires a linear number of messages, similarly to DPOP's Utility Propagation phase. Additionally, the *Value Propagation* phase requires a linear number of messages, as in DPOP. The updated value ω is propagated down the pseudo-tree during the value propagation phase. Each agent can thus start the next iteration immediately after completing its value propagation step. □

Let $\ell = \max_{a_i \in \mathcal{A}} |D_i|^{H_{opt}}$ be the largest size of the initial utility table of the agents at each iteration.

Theorem 2. *The size of the largest message exchanged by the* DEEDS *agents is* $O(\ell^{|\mathcal{A}|})$.

Proof. Due to the inability of the agents to bound the flow values associated to a particular net injection—which in turn is caused by the presence of the global constraint on the line flows (Eq. (9))—the aggregation of k tables of size $O(\ell)$ produces a new table of size $O(\ell^k)$. The number of aggregation steps performed by all DEEDS agents is $|\mathcal{A}|$ (during the *Utility Propagation* step, each agent aggregates the Utility tables received from its children). Thus, the size of the largest message processed at each iteration is $O(\ell^{|\mathcal{A}|})$. □

Theorem 3. *The size of the largest message exchanged by the R-*DEEDS *agents is* $O(|\mathcal{A}| \cdot \ell)$.

Proof. During the aggregation step, the number of rows of the table resulting by aggregating tables for k R-DEEDS agents $a_1, \ldots, a_k$ is: $\left[(\sum_{i=1}^{k} G_{a_i}^{max} - \sum_{i=1}^{k} G_{a_i}^{min}) \cdot (\sum_{i=1}^{k} L_{a_i}^{max} - \sum_{i=1}^{k} L_{a_i}^{min})\right]^{H_{opt}}$, which is in $O(k \cdot \ell)$. Since R-DEEDS agents perform in total $|\mathcal{A}|$ aggregation operations per iteration, the size of the largest message processed at each iteration is $O(|\mathcal{A}| \cdot \ell)$. □

8. RELATED WORK

Within the power systems literature, the ED and the DR solutions are typically implemented separately despite the

IEEE Buses	H_{opt}	Simulated Runtime (sec) CPU Implementation 1	CPU 2	CPU 3	CPU 4	GPU Implementation 1	GPU 2	GPU 3	GPU 4	Normalized Quality 1	Q 2	Q 3	Q 4
	5	0.010	0.044	3.44	127.5	0.025 (0.4x)	0.038 (1.2x)	0.128 (26.9x)	2.12 (60.2x)	0.8732	0.8760	0.9569	1.00
	14	0.103	509.7	–	–	0.077 (1.3x)	3.920 (130x)	61.70 (n/a)	–	0.6766	0.8334	1.00	–
	30	0.575	9084	–	–	0.241 (2.4x)	79.51 (114x)	–	–	0.8156	1.00	–	–
	57	4.301	–	–	–	0.676 (6.4x)	585.4 (n/a)	–	–	0.8135	1.00	–	–
	118	174.4	–	–	–	4.971 (35.1x)	–	–	–	1.00	–	–	–

Table 1: R-Deeds CPU and GPU Runtimes, Speedups (in parenthesis), and Normalized Solution Qualities.

strong coupling of the two problems [7]. While some researchers have more recently proposed fully distributed ED solutions as well as an integrated ED with DR solutions [41, 18], their approaches do not consider a time horizon, and are limited to convex objective functions. In conventional ED, the objective function is typically assumed to be piecewise linear or quadratic, differentiable, and convex [37]. In this paper, we consider non-convex, non-smooth objective functions, which more realistically reflect the valve-point effects and the POZs. We also note that non-linearities arise in power flow analysis when AC models are introduced. Recently, several studies addressing various convex AC relaxations have been proposed [5, 6]. There are other proposals on non-convex objective functions [29, 40]; however, such approaches are centralized and consider exclusively the ED problem. An exception is given in [28], where the authors study a distributed DR mechanism in a microgrid setting (within a convex optimization context).

Finally, to the best of our knowledge, we are the first to propose an integrated model with a time horizon. Most approaches are myopic and only considers a single time step. The time horizon is important as it allows solutions that consider ramp constraints—a large generator can typically change its generation up to a *maximum rate limit* of several MWs per minute. If the solution to the ED problem requires the generator to change beyond this maximum rate limit, then it must take several time steps to do so, possibly resulting in suboptimal generation at the intermediate time steps. This solution is only feasible if the model includes a time horizon.

Within the AI community, researchers have also studied variants of the ED problem including several that use DCOP-based approaches [19, 16, 12] and one that uses a negotiation-based approach [39]. The main differences with our work is that their proposals simplify the energy constraints by assuming that agents have direct discrete control over the power flows, and the problems are solved for a single time step, thereby not taking into account ramp rate constraints. AI researchers have also studied variants of the DR problem including a proposal to use a centralized online stochastic optimization approach for a home automation system [27]. Scott and Thiébaux [28] and Fioretto *et al.* [10] have also studied distributed DR mechanism for the scheduling of shiftable loads in smart homes.To the best of our knowledge, a combined ED and DR multi-timestep framework is unique within the AI literature.

9. EMPIRICAL EVALUATION

We evaluate the proposed algorithms on 5 standard IEEE Bus Systems,[7] all having non-convex solution spaces:

- **IEEE 5-Bus System:** 1 generator, 5 loads, and 7 transmission lines [17].
- **IEEE 14-Bus System:** 5 generators, 11 loads, and 20 transmission lines [36].
- **IEEE 30-Bus System:** 6 generators, 27 loads, and 41 transmission lines [1].
- **IEEE 57-Bus System:** 7 generators, 42 loads, and 80 transmission lines (as well as multiple transformers) [26].
- **IEEE 118-Bus System:** 54 generators, 91 loads, and 177 transmission lines (as well as multiple condensers and transformers) [3].

The domains for generator and load variables, as well as the transmission lines' admittances and maximum flows, and the cost coefficients for the generators are defined in the associated references. The coefficients ϵ_g, ϕ_g, are defined as in [29]. We set one 10MW range POZ to a single generator in each of the bus systems analyzed, similar to [11]. All the generators are set with a 5MW ramp rate limit. We use a 1MW discretization unit for each load and generators. Thus, the maximum domain sizes of the variables are between 100 and 320 in every experiment. For each test system, the loads' cost coefficient α_l and β_l are randomly sampled from the normal distributions $\mathcal{N}(1, 0.1)$, and $\mathcal{N}(0.01, 0.001)$, respectively, and the horizon H is set to 12.

We generate 30 instances for each test case. The performance of the algorithms are evaluated using the *simulated runtime* metric [30], and we imposed a timeout of 5 hours. Results are averaged over all instances. These experiment are performed on an *AMD Opteron 6276*, 2.3GHz, 128GB of RAM, which is equipped with a *GeForce GTX TITAN* GPU device with 6GB of global memory. If an instance cannot be completed, it is due to runtime limits for the CPU implementation of the algorithm, and memory limits for the GPU implementation. We set the parameters for the iterative algorithm $m = 0.9$ and $\omega = 100$, as is done in [18], and the number of iterations to 10. Automating this process is the objective of future work.

We do not report evaluation against existing (Dynamic) DCOP algorithms as the standard (Dynamic) DCOP models cannot fully capture the complexities of the D-EDDR problem (e.g., environment variables with continuous values).

Table 1 tabulates the runtimes in seconds of R-Deeds with both CPU and GPU implementations at varying H_{opt}. It also tabulates the average solution quality found normalized with the best solution found for each IEEE bus. We make the following observations:

- The solution quality increases as H_{opt} increases.
- For the smaller test case with $H_{opt} = 1$, the CPU implementation is faster than the GPU one. However, for all other configurations, the GPU implementation is much faster than its CPU counterpart, with speedups up to

[7] http://publish.illinois.edu/smartergrid/

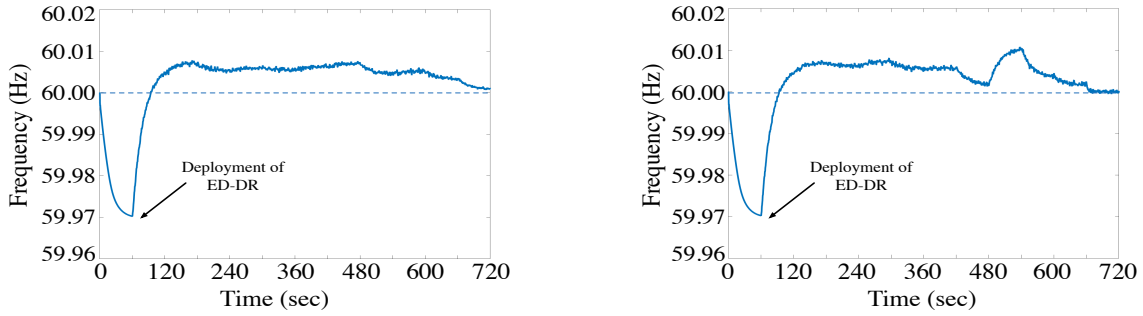

Figure 5: Dynamic Simulation of the IEEE 30 Bus with heavy loads with high (left) and low (right) prediction probabilities.

130 times. The reported speedup increase with increasing complexity of the problem (i.e., bus size and H_{opt}).

- The GPU implementation scales better than the CPU implementation. The latter could not solve any of the instances for the configurations with $H_{opt} = 3$ for the 14-bus system and $H_{opt} = 2$ for the 57-bus system.
- We observe that the algorithms report satisfiable instances within the first 4 iteration of the iterative process.

9.1 Dynamic Simulation on a State-of-the-Art Power Systems Simulator

To validate the solutions returned by R-DEEDS, we tested the stability of the returned control strategy for the IEEE 30-Bus System on a dynamic simulator (Matlab *SimPowerSystems*) that employs a detailed time-domain model and recreates conditions analogous to those of physical systems.

An illustration of our *Matlab Simulink* model for the IEEE 30 Bus System is shown on the left-hand side of Figure 1. The right-hand side of the figure also depicts the implementation of a generator and a dispatchable load.

We created the following scenarios:

- In the first scenario, load prediction probabilities are assumed to be accurate. We perturb each load l_t using a normal distribution $\mathcal{N}(\mu, \sigma)$ with $\mu = L_l^t$ and $\sigma = (1 - \alpha^t) \cdot \sqrt{\mu}$.
- In the second scenario, load predictions are less accurate and we use a larger variance $\sigma = (1 - \alpha^t) \cdot 2\sqrt{\mu}$.

Figure 5 shows the dynamic response of the system frequency, whose nominal value is 60 Hz, for the two scenarios. The system frequency is at nominal value when the power supply-demand is balanced. If more power is produced than consumed, the frequency would rise and vice versa. Deviations from the nominal frequency value would damage synchronous machines and other appliances. Thus, to ensure stable operating conditions, it is important to ensure that the system frequency deviation is confined to be within a small interval (typically 0.1 Hz).

In our experiment, the first 60 seconds of simulations are tested enforcing the ED solution in a full load scenario (100% of full load) using the same setting as in [18]. The rest of the simulation deploys the ED-DR solution returned by R-DEEDS. While the resolution of the simulation is in microseconds, R-DEEDS agents sends only desired power injected and withdrawn (schedules), computed offline, in one-minute intervals; the simulator interpolates the power generated between the intervals. This scenario reflects real-life conditions, where control signals are sent to actual generators in intervals of several minutes. As expected, the frequency deviation is more stable when the load predictions are accurate. Crucially, the deviation in both cases is within 0.05 Hz, thereby ensuring system stability. In addition, we observe, during simulation, that the ED-DR solution is able to reduce the total load up to 68.5%, showing the DR contribution in peak demand reduction. Finally, the frequency response of the ED-DR solution converges faster than that of the ED only, with smaller fluctuations. The reason is because the supply-demand balance is better maintained by coordinating simultaneously the generators and loads in the system.

10. CONCLUSIONS

Responding to the recent call to action for AI researchers to contribute towards the smart grid vision [25], this paper makes three key contributions: (1) An integrated ED and DR model for power systems, (2) a multi-agent based approach to solve it which exploits structural problem decomposition, and (3) a GPU parallelization of its operations. This model differs from existing work in the power systems and AI literature primarily in its multi-timestep optimization problem, which is necessary to capture ramp constraints, as well as in its more realistic non-convex objective function. The decentralized DCOP-based algorithm solves a relaxed version of the integrated model and achieves significant speedup through the use of GPUs on IEEE buses of varying sizes. The proposed results are significant—evaluations on a state-of-the-art power systems simulator show that the solutions found are stable with acceptable frequency deviations. Therefore, this work continues to pave the bridge between the AI and power systems communities, highlighting the strengths and applicability of AI techniques in solving power system problems. Future work will focus on scalability and on computing solutions using an iterative process to refine the level of discretization for the generators and load outputs.

Acknowledgments

This research is partially supported by NSF grants 1345232 and 0947465. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the sponsoring organizations, agencies, or the U.S. government.

REFERENCES

[1] O. Alsac and B. Stott. Optimal load flow with steady-state security. *IEEE Transaction on Power Systems*, PAS-93(3):745–751, 1974.

[2] G. Binetti, A. Davoudi, D. Naso, B. Turchiano, and F. L. Lewis. A distributed auction-based algorithm for the nonconvex economic dispatch problem. *IEEE Transaction on Industrial Informatics*, 10(2):1124–1132, 2014.

[3] S. Blumsack. *Network topologies and transmission investment under electric-industry restructuring*. ProQuest, 2006.

[4] B. H. Chowdhury and S. Rahman. A review of recent advances in economic dispatch. *IEEE Transactions on Power Systems*, 5(4):1248–1259, 1990.

[5] C. Coffrin and P. V. Hentenryck. A linear-programming approximation of AC power flows. *INFORMS Journal on Computing*, 26(4):718–734, 2014.

[6] C. Coffrin, H. L. Hijazi, and P. V. Hentenryck. The QC relaxation: Theoretical and computational results on optimal power flow. *CoRR*, abs/1502.07847, 2015.

[7] A. J. Conejo, J. M. Morales, and L. Baringo. Real-time demand response model. *IEEE Transaction on Smart Grid*, 1(3):236–242, 2010.

[8] R. Deng, Z. Yang, J. Chen, N. Asr, and M. Chow. Residential energy consumption scheduling: A coupled-constraint game approach. *IEEE Transactions on Smart Grid*, 5(3):1340–1350, 2014.

[9] F. Fioretto, T. Le, W. Yeoh, E. Pontelli, and T. C. Son. Exploiting GPUs in solving (distributed) constraint optimization problems with dynamic programming. In *Proc. of CP*, pages 121–139, 2015.

[10] F. Fioretto, W. Yeoh, and E. Pontelli. A multiagent system approach to scheduling devices in smart homes. In *Proc. of AAMAS*, 2017.

[11] Z. Gaing. Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Transaction on Power Systems*, 18(3):1187–1195, 2003.

[12] S. Gupta, W. Yeoh, E. Pontelli, P. Jain, and S. J. Ranade. Modeling microgrid islanding problems as DCOPs. In *Proc. of NAPS*, pages 1–6, 2013.

[13] Y. Hamadi, C. Bessière, and J. Quinqueton. Distributed intelligent backtracking. In *Proc. of ECAI*, pages 219–223, 1998.

[14] H. Johal, K. Anaparthi, and B. Jason. Demand response as a strategy to support grid operation in different time scales. In *Proc. of ECCE*, pages 1461–1467, 2012.

[15] J. K. Kok, C. J. Warmer, and I. Kamphuis. Powermatcher: multiagent control in the electricity infrastructure. In *Proc. of AAMAS*, pages 75–82, 2005.

[16] A. Kumar, B. Faltings, and A. Petcu. Distributed constraint optimization with structured resource constraints. In *Proc. of AAMAS*, pages 923–930, 2009.

[17] F. Li and R. Bo. Small test systems for power system economic studies. In *Proc. of IEEE PES General Meeting*, pages 1–4, 2010.

[18] Y. Ma, W. Zhang, W. Liu, and Q. Yang. Fully distributed social welfare optimization with line flow constraint consideration. *IEEE Transaction on Industrial Informatics*, 11(6):1532–1541, 2015.

[19] S. Miller, S. Ramchurn, and A. Rogers. Optimal decentralised dispatch of embedded generation in the smart grid. In *Proc. of AAMAS*, pages 281–288, 2012.

[20] P. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1–2):149–180, 2005.

[21] H. Nunna and S. Doolla. Multiagent-based distributed-energy-resource management for intelligent microgrids. *IEEE Transaction on Industrial Electronics*, 60(4):1678–1687, 2013.

[22] P. Palensky and D. Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *IEEE Transaction on Industrial Informatics*, 7(3):381–388, 2011.

[23] A. Petcu and B. Faltings. A scalable method for multiagent constraint optimization. In *Proc. of IJCAI*, pages 1413–1420, 2005.

[24] A. Petcu and B. Faltings. Superstabilizing, fault-containing distributed combinatorial optimization. In *Proc. of AAAI*, pages 449–454, 2005.

[25] S. D. Ramchurn, P. Vytelingum, A. Rogers, and N. R. Jennings. Putting the 'smarts' into the smart grid: A grand challenge for artificial intelligence. *Communications of the ACM*, 55(4):86–97, 2012.

[26] W. Rosehart, C. Canizares, and V. Quintana. Multiobjective optimal power flows to evaluate voltage security costs in power networks. *IEEE Transaction on Power Systems*, 18(2):578–587, 2003.

[27] P. Scott, S. Thiébaux, M. van den Briel, and P. van Hentenryck. Residential demand response under uncertainty. In *Proc. of CP*, pages 645–660, 2013.

[28] P. Scott and W. Thiébaux. Distributed multi-period optimal power flow for demand response in microgrids. In *Proc. of e-Energy*, pages 17–26, 2015.

[29] A. Selvakumar and K. Thanushkodi. A new particle swarm optimization solution to nonconvex economic dispatch problems. *IEEE Transaction on Power Systems*, 22(1):42–51, 2007.

[30] E. Sultanik, P. J. Modi, and W. C. Regli. On Modeling Multiagent Task Scheduling as a Distributed Constraint Optimization Problem. In *Proc. of IJCAI*, pages 1531–1536, 2007.

[31] D. Sun *et al.* Optimal power flow by newton approach. *IEEE Transaction on Power Apparatus and Systems*, PER-4(10):2864–2880, 1984.

[32] D. Todd, M. Caufield, B. Helms, A. P. Generating, I. M. Starke, B. Kirby, and J. Kueck. Providing Reliability Services through Demand Response: A Preliminary Evaluation of the Demand Response Capabilities of Alcoa Inc. Oak Ridge National Laboratory, 2009.

[33] J. Tong and H. Ni. Look-ahead multi-time frame generator control and dispatch method in PJM real time operations. In *Proc. of IEEE PES General Meeting*, pages 1–1, 2011.

[34] D. Walters and G. Sheble. Genetic algorithm solution of economic dispatch with valve point loading. *IEEE Transaction on Power Systems*, 8(3):1325–1332, 1993.

[35] S. Wang, S. Shahidehpour, D. Kirschen, S. Mokhtari, and G. Irisarri. Short-term generation scheduling with transmission and environmental constraints using an augmented lagrangian relaxation. *IEEE Transaction on Power Systems*, 10(3):1294–1301, 1995.

[36] P. Wong *et al.* The IEEE reliability test system-1996. *IEEE Transaction on Power Systems*, 14(3):1010–1020, 1999.

[37] A. J. Wood and B. F. Wollenberg. *Power generation, operation, and control*. John Wiley & Sons, 2012.

[38] L. Xie, P. M. S. Carvalho, L. A. F. M. Ferreira, J. Liu, B. H. Krogh, N. Popli, and M. D. Ilic. Wind integration in power systems: Operational challenges and possible solutions. *Proc. of the IEEE*, 99(1):214–232, 2011.

[39] D. Ye, M. Zhang, and D. Sutanto. Decentralised dispatch of distributed energy resources in smart grids via multi-agent coalition formation. *Journal of Parallel and Distributed Computing*, 2015.

[40] J. Zhan, Q. Wu, C. Guo, and X. Zhou. Economic dispatch with non-smooth objectives part ii: Dimensional steepest decline method. *IEEE Transaction on Power Systems*, 30(2):722–733, 2015.

[41] W. Zhang, W. Liu, X. Wang, L. Liu, and F. Ferrese. Online optimal generation control based on constrained distributed gradient algorithm. *IEEE Transaction on Power Systems*, 30(1):35–45, 2015.