Energy Syst (2012) 3:153–179
DOI 10.1007/s12667-012-0050-4

ORIGINAL PAPER

An energy management system for off-grid power systems

Daniel Zelazo · Ran Dai · Mehran Mesbahi

Received: 3 November 2011 / Accepted: 16 January 2012 / Published online: 31 January 2012

Abstract Next generation power management at all scales will rely on the efficient scheduling and operation of both generating units and loads to maximize efficiency and utility. The ability to schedule and modulate the demand levels of a subset of loads within a power system can lead to more efficient use of the generating units. These methods become increasingly important for systems that operate independently of the main utility, such as microgrid and off-grid systems. This work extends the principles of unit commitment and economic dispatch problems to off-grid power systems where the loads are also schedulable. We propose a general optimization framework for solving the energy management problem in these systems. An important contribution is the description of how a wide range of sources and loads, including those with discrete states, non-convex, and nonlinear cost or utility functions, can be reformulated as a convex optimization problem using, for example, a shortest path description. Once cast in this way, solution are obtainable using a subgradient algorithm that also lends itself to a distributed implementation. The methods are demonstrated by a simulation of an off-grid solar powered community.

Keywords Off-grid · Energy management · Lagrangian relaxation · Shortest-path algorithm

D. Zelazo (✉)
Institute for Systems Theory and Automatic Control, Universität Stuttgart, 70550 Stuttgart, Germany
e-mail: daniel.zelazo@ist.uni-stuttgart.de

R. Dai · M. Mesbahi
Department of Aeronautics and Astronautics, University of Washington, Seattle, WA, 98195-2400, USA

R. Dai
e-mail: dairan@washington.edu

M. Mesbahi
e-mail: mesbahi@washington.edu

Nomenclature

G, L	number of generating units, number of loads
$\mathcal{G}, \mathcal{L}$	set of generating units, set of loads
T	time horizon
$\mathcal{T}$	set of time indices
i, j, t	index for generating units, loads, and time
$g_i(t), y_j(t)$	power level of generating unit $i \in \mathcal{G}$ and demand of load $j \in \mathcal{L}$ at time $t \in \mathcal{T}$
$x_i^g(t), u_i^g(t)$	state and control variables for generating unit $i \in \mathcal{G}$ at time $t \in \mathcal{T}$
$x_j^l(t), u_j^l(t)$	state and control variables for load $j \in \mathcal{L}$ at time $t \in \mathcal{T}$
$f_i^g(g_i(t), x_i^g(t), u_i^g(t))$	dynamic evolution of generating unit variables for unit $i \in \mathcal{G}$
$f_j^l(y_j(t), x_j^l(t), u_j^l(t))$	dynamic evolution of load variables for load $j \in \mathcal{L}$
$C_i(g_i(t), x_i^g(t), u_i^g(t))$	operating cost of generator unit $i \in \mathcal{G}$ at time $t \in \mathcal{T}$
$U_j(y_j(t), x_j^l(t), u_j^l(t))$	utility of load $j \in \mathcal{L}$ at time $t \in \mathcal{T}$
$\mathcal{S}_i^g(t), \mathcal{S}_j^l(t)$	abstract constraint set for generator unit $i \in \mathcal{G}$ and load $j \in \mathcal{L}$ at time $t \in \mathcal{T}$
$\mathcal{X}_i^g(t), \mathcal{U}_i^g(t)$	abstract constraint set for generator load state and control variables at $t \in \mathcal{T}$
$\mathcal{X}_j^l(t), \mathcal{U}_j^l(t)$	abstract constraint set for load state and control variables at $t \in \mathcal{T}$
$\mathbf{L}(\cdot), q(\lambda)$	Lagrangian and dual functions
$\lambda_t, v(\lambda_t), \alpha$	Lagrange multiplier and sub-gradient at time $t \in \mathcal{T}$, step size

1 Introduction

The push to modernize power systems towards a "smart" grid is driven by a variety of factors including environmental compliance, economic advantage, and improved reliability, robustness, and service. Indeed, the development of technologies and applications related to smart grids is now a centerpiece for a clean-energy economy and initiative in both the United States and European Union [1, 2]. Precisely how a smart grid should look like and operate is now an active area of research in the power systems community [3–6].

The ultimate success of a smart-grid infrastructure will depend on a decreasing dependence of consumers on the main power distribution network. A first step in this direction is through the so-called *microgrid*. A *microgrid* contains various distributed energy resources (i.e., wind turbines, photovoltaic arrays, fuel cells, generators), energy storage devices (i.e., super-capacitors, batteries), and controllable loads [7]. A key feature of microgrids is their ability to operate autonomously, i.e. isolated from the main power grid. The success of a microgrid-type architecture depends not only on the development of new energy resources and storage capabilities, but equally on the ability to control and schedule these systems in a distributed manner [8]. Another challenge associated with microgrid is managing the import and

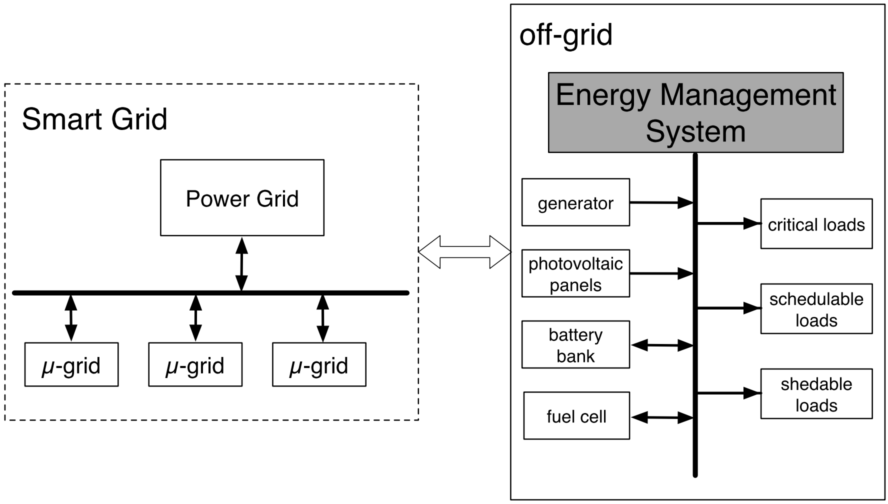


Fig. 1 Off-grid systems operate independently of a microgrid and smart grid architecture

export of power to the main distribution network. This relates to both the power balance of the entire network and the economic bartering between the utility and the microgrid [9, 10].

While much of the vision for smart and microgrids are aimed at large scale implementation, many of the fundamental principles and techniques can also be adapted for smaller-sized self-contained power systems. Such systems are referred to as *off-grid* systems [11–13]. Off-grid systems inherit many similarities to microgrids, primarily regarding the ability to operate independently of the main grid; this is visualized in Fig. 1. Off-grid systems can include small self-contained power systems such as hybrid and electrical vehicles (HEV) [14–16] and the "more electric aircraft" (MEA) pursued by the United States Air Force [17–19]. A larger scale version, including the scenario examined in this work, is that of smart homes and communities residing off the main grid [10, 20, 21].

The main component of an off-grid system, therefore, is an effective *energy management system* (see Fig. 1). The energy management system must coordinate the scheduling and operation of all the distributed generation sources in the system while also managing the efficient use of the loads. This should be done in a way that the overall operating cost for generation is minimized and the utility obtained by the user through the loads is maximized. Traditionally, the scheduling and operation of large scale power systems involves the solution of the unit commitment (UC) and economic dispatch (ED) problems. These problems determine an optimal schedule and commitment level for each generating unit in a power system based on a set of constraints, including reserve power, operating parameters, and forecasted loads over a finite time horizon. Many solution methods have been proposed including Lagrangian relaxation techniques and dynamic programming [22–26]. The mixed-integer linear programming (MILP) method applicable to small and medium sized power systems with linear models is another approach to search for the optimal continuous and integer variables [27–29]. One of the difficulties related to this class of problems, however, is that the demand level is often not known precisely.

While this issue is perhaps unavoidable when considering large scale power systems, the notion of unknown load demand levels for smaller scale power systems may not be as prevalent. In applications such as HEV and MEA, certain classes of loads may follow very precise profiles for their demand level. Such loads, for example, avionics in an aircraft or a refrigerator in an off-grid home, may be critical to the successful operation of the power system, while others, such as the environmental control system, may follow looser demand profiles and have flexibility in their demand level. In this direction, we consider an extension of the economic dispatch and unit commitment problem to include the scheduling and operation of a subset of loads within the power system. The notion of an optimal load demand pattern can be considered as the "dual" problem to the ED and UC formulation; if generation of power is modeled as a fixed resource then indeed the scheduling of loads would follow the same methodology. For ED and UC of power generation, the optimization objective is the minimization of the overall fuel and operating costs. For loads, we consider the maximization of the load utility. The utility of certain loads is expected to be highly dependent on the application, and the exact form of these utility functions may not have the same analytic foundation as for generation. A possible source for these utility functions might be borrowed from microeconomic principles in the form of a *demand curve* or *benefit function* [30]. Such utility-based approaches have been considered for real-time pricing algorithms within a smart grid framework with quadratic or linear utility functions [31, 32]. In [33], an energy management system for smart homes aimed at maximizing the net benefit by end users is considered. The general problem of utility maximization has been studied for problems related to network architecture and resource allocation [34, 35].

The notion of scheduling loads in a power system is not an entirely new idea. A common technique employed in self-contained power systems is *load shedding*. In many situations, such as in aircraft power systems, load shedding is based on a heuristic where loads are "ranked" according to a priority list [36, 37]; when there is more demand than power available, loads at the bottom of the list are shed until the power balance is restored. More sophisticated load shedding methods utilizing techniques from optimization have also been employed [38], but the fundamental notion of a priority is still utilized. Direct load control (DLC) is another attempt to control the load demand although the general approach is motivated by the capabilities of the generating units rather than the utility of the loads [39–41]. With the recognition of the importance of smart grids, DLC has been revisited with a broader focus but is still faced with challenges such as the understanding the relationship between the consumer and the supplier [30, 42–44].

The main contribution of this work, therefore, is the development of an energy management system for an off-grid power system. This work extends the principles and techniques employed in traditional UC and ED problems to self-contained power systems that also permit the scheduling of loads. Specifically, we formulate an optimization problem that minimizes the operating and generating costs of a distributed generation system while simultaneously maximizing the utility of the loads in the power system. In the process of deriving the optimization model, we describe how qualitative descriptions of generating units and loads can be mathematically modeled in a way that leads to tractable solution algorithms for solving the problem; this greatly generalizes the current methods used related to load control in off-grid systems. The structure of the proposed model lends itself to a distributed implementation using a sub-gradient algorithm. An important component of the algorithm described here is the use of a shortest path solution to solve component level sub-problems based on the qualitative description of each unit. The proposed distributed algorithm allows for parallel computation while maintaining stable calculation performance with increasing number of electric components in the system. Furthermore, the qualitative descriptions developed allows to consider a broader class of utility functions, including discrete, nonlinear, and non-convex (such as a priority-type utility), as compared to other solutions [31, 32]. Finally, the algorithm is easy to implement, leading to a cheaper solution compared to that of commercial MILP software. A detailed discussion of the advantages of a Lagrangian relaxation technique compared to MILP is given in [28, 29].

The outline of this paper is as follows. In Sect. 2 we first develop the power system model employed in this work. This section includes a generic description of the generating units and loads including their cost and utility functions along with their associated constraints. The optimization model is also derived in Sect. 2 along with the companion dual problem. The description of the dual problem provides insights into alternative descriptions of the model, that in turn, leads to tractable solution methods. In Sect. 2.1 we explore how a wide description of generating units and loads can be formulated into equivalent tractable problems. The description of a decentralized solution method for solving the model is presented in Sect. 3. A simulation example of an off-grid solar powered community is provided in Sect. 4, and concluding remarks are given in Sect. 5.

2 An optimization framework for an energy management system

The general objective for a distributed generation system with schedulable loads is to determine a schedule and commitment level for each generating unit and load such that the running costs of the generating units are minimized and the utility of the loads are maximized.[1] In addition to cost minimization and utility maximization, each unit must also respect its own operating constraints and the aggregate power balance of the system.

Generally, the generating units and loads can contain systems with both continuous and discrete operating states. For example, a battery might be modeled to have a constant charge and discharge rate. In this way, the variable $g_i(t)$, corresponding to the battery power level, will be a discrete variable. In contrast, a generator can output power continuously, corresponding to $g_i(t)$ being a continuous variable. The continuous or discrete nature of each element is encoded in the corresponding abstract constraint sets for the units (described in the nomenclature section).

[1] At times we will also refer to the "disutility" of loads. In this way, we aim to minimize the disutility.

In this direction, we can express the objective of the energy management system that we would like to minimize as,

$$J(\mathbf{g},\mathbf{y},\mathbf{x}^g,\mathbf{x}^l,\mathbf{u}^g,\mathbf{u}^l) = \sum_{t=1}^{T}\left(\sum_{i=1}^{G} C_i\big(g_i(t), x_i^g(t), u_i^g(t)\big) - \sum_{j=1}^{L} U_j\big(y_j(t), x_j^l(t), u_j^l(t)\big)\right). \tag{2.1}$$

For notational simplicity we introduced the bold-faced vector notation; for example $\mathbf{g}(t) = [\, g_1(t) \cdots g_G(t)\,]^T$.

Remark 1 The cost minimization or utility maximization component of the objective function (2.1) can be assigned a "priority" by introducing a scalar weight for each term. For example, assigning a weight of 0 to the utility maximization term corresponds to the cost function for a traditional ED and UC problem.

When considering an optimization framework for solving the scheduling and commitment problems, it is standard to assume certain properties for the objective function (2.1). For example, convexity and continuity of the cost function (and concavity of the utility function) allow, through the tools of convex analysis, efficient solution methods in addition to strong guarantees on rates of convergence and optimality [45]. However, in many qualitative descriptions of generating units and loads, such an ideal mathematical description of their cost or utility might not be possible. Rather, the functions may have discontinuities and convexity or linearity might not be guaranteed. Combined with the possibly discrete nature of the optimization variables, minimization of (2.1) becomes computationally difficult. We will show in this section that for certain forms of the cost and utility functions, equivalent representation of the objective can be derived that lead to tractable solution methods.

The optimization problem we aim to solve, which we call problem $\mathcal{P}$, can now be stated as

$$\min_{\mathbf{g},\mathbf{y},\mathbf{x}^g,\mathbf{x}^l,\mathbf{u}^g,\mathbf{u}^l} \quad J(\mathbf{g},\mathbf{y},\mathbf{x}^g,\mathbf{x}^l,\mathbf{u}^g,\mathbf{u}^l) \tag{2.2}$$

$$\text{s.t.} \quad (g_i(t), x_i^g(t), u_i^g(t)) \in \mathcal{Q}_i^g(t), \quad \forall i \in \mathcal{G}, t \in \mathcal{T} \tag{2.3}$$

$$(y_j(t), x_j^l(t), u_j^l(t)) \in \mathcal{Q}_j^l(t), \quad \forall j \in \mathcal{L}, t \in \mathcal{T} \tag{2.4}$$

$$x_i^g(t+1) = f_i^g(g_i(t), x_i^g(t), u_i^g(t)), \quad \forall i \in \mathcal{G}, j \in \mathcal{L} \tag{2.5}$$

$$x_j^l(t+1) = f_j^l(y_j(t), x_j^l(t), u_j^l(t)), \quad \forall i \in \mathcal{G}, j \in \mathcal{L} \tag{2.6}$$

$$\sum_{j=1}^{L} y_j(t) = \sum_{i=1}^{G} g_i(t), \quad \forall t \in \mathcal{T}. \tag{2.7}$$

We introduced a further notation for simplification, $\mathcal{Q}_i^g(t) = \mathcal{S}_i^g(t) \times \mathcal{X}_i^g(t) \times \mathcal{U}_i^g(t)$ and $\mathcal{Q}_i^l(t) = \mathcal{S}_i^l(t) \times \mathcal{X}_i^l(t) \times \mathcal{U}_i^l(t)$. The constraints (2.5) and (2.6) represent any dynamic descriptions of a unit, such as the decision to turn a unit on or off. The constraint (2.7) is the aggregate power balance equation of the power system. Note that this constraint is the only coupling constraint in the problem $\mathcal{P}$; in the absence of such a constraint the problem could be decomposed in a straightforward way, and much of the challenge in solving $\mathcal{P}$ stems from this constraint. It is also worth mentioning that the power balance equation (2.7) assumes that the power system resides on a single power bus.

The dimension of this program is dependent on the time horizon, $T = |\mathcal{T}|$ and the number of generating units G and loads L. In this regard, the size of this problem is $\mathcal{O}(T(G + L))$; the exact size will depend on the complexity of the dynamic and abstract constraints for each generating unit and load. Finally, we would like to re-emphasize that the model in (2.2), in its most general form, represents a *non-linear*, *non-convex*, and *mixed-integer* program.

Remark 2 The framework presented above provides, in general, a complete schedule and commitment level for all units in the power system. However, certain scenarios might require that a unit follow a nominal or desired state trajectory. Our earlier discussion also handles the situation where a nominal trajectory must be satisfied (for example, by specifying the constraint sets $\mathcal{S}^g$ and $\mathcal{S}^l$ as an equality constraint at each time t). We also would like to highlight that our problem formulation implicitly handles a scenario where a nominal trajectory must be tracked, but the unit is allowed to deviate from that trajectory in some specified way. In particular, we note that the cost functions for each unit can be designed such that the error between a nominal trajectory and the real trajectory is minimized (e.g., $C_i(g_i(t)) = (g_i(t) - \overline{g}(t))^2$ for some reference trajectory $\overline{g}(t)$). The constraint sets $\mathcal{S}^g$ and $\mathcal{S}^l$ would then capture the allowable deviations from the nominal trajectory (e.g., $g_i(t) \in [\overline{g}(t) - \epsilon, \overline{g}(t) + \epsilon]$).

Before considering solution methods for the problem $\mathcal{P}$, we first describe a few classes of units and their associated objective functions and constraint sets. We would like to emphasize that the precise form of these functions is of paramount importance for the development of a solution methodology.

2.1 Unit descriptions

A significant challenge in solving problem $\mathcal{P}$ is a precise description of the cost and utility functions, and the constraints of each unit (both dynamic and static). In this direction, we define here a few general classes of units.

2.1.1 Convex and continuous objective, 1 state (CC1)

This represents the simplest class of a generating unit or load. We assume the cost (utility) function is a convex (concave) and continuous function. While the general form of the cost function (e.g. piecewise continuous or smooth) will ultimately dictate the appropriate solution method, this form allows for a traditional approach for finding a solution method. For example, one might consider a piecewise-linear cost curve representing a progressively increasing cost as the output power increases. An example of a utility function for loads could be an α-fair utility curve commonly used in network utility maximization problems [34].

The other defining property for units of this type relates to the number of states required to describe its operation. Units that are 'always on' do not have to be scheduled as there is no ability to change the state of the unit; for example, a refrigerator in an off-grid home should be modeled as such a unit. Critical loads, such as avionics in aircraft, can also be modeled in this way. Therefore, the dynamic constraints (2.5) and (2.6) are not needed.[2] The only constraints these units must consider are the abstract constraint sets $\mathcal{S}_i^g(t)$ and $\mathcal{S}_j^l(t)$ corresponding to the lower and upper bounds on the power output (demand) level of the unit. In the absence of a dynamic constraint, the objective from the perspective of each unit is to minimize the cost of generation (maximize the load utility) while respecting the aggregate power balance constraint of the system.

2.1.2 Convex and continuous objective, n states (CCn)

A natural generalization of the class *CC1* is to include multiple operating states describing the operation of the unit. As with *CC1*, we assume the cost and utility functions are convex (concave) and continuous. The addition of multiple operating states adds the scheduling component to problem $\mathcal{P}$. This class of units perhaps describes the most common types of units found in a self-contained power system. While the exact nature of each unit will require a customized model, we will show here a few examples to illustrate the general procedure. The following models represent extensions and variations to the basic models used in [22, 26].

Example 1 (On/Off units) The simplest example is for $n = 2$, corresponding to a unit that can be switched on or off. This type of unit has dynamics of the form[3]

$$x^g(t+1) = u^g(t), \tag{2.8}$$

with $x(t) \in \mathcal{X} = \{0, 1\}$ and $u(t) \in \mathcal{U} = \{0, 1\}$, for all $t \in \mathcal{T}$. Here we associate the state value $x(t) = 0$ to 'off' and $x(t) = 1$ to 'on.' Using this dynamic description we are also able to explicitly consider any costs associated with a state transition. For example, the cost function $C_i(g(t), x^g(t), u^g(t))$ (and utility $U_i(y(t), x^l(t), u^l(t))$) given in (2.1) can be decomposed to reflect transition costs in addition to generation (utility) costs. In this way, we can write the cost function as

$$C\big(g(t), x^g(t), u^g(t)\big) = P^g(g(t), x^g(t)) + S^g(x^g(t), u^g(t)); \tag{2.9}$$

the function is composed of a cost associated with generating power at a level $g(t)$ (and the state to ensure that when the device is 'off' there is no cost for generation),

[2] They can be included for completeness by defining, for example, $x_i^g(t+1) = f_i^g(g_i(t), x_i^g(t), u_i^g(t)) = 0$.

[3] A similar formulation can be used for the loads.

and the cost of switching states in the next time interval.[4] The associated cost for the load will have the similar form;

$$U\big(y(t), x^l(t), u^l(t)\big) = P^l(y(t), x^l(t)) - S^l(x^l(t), u^l(t)); \tag{2.10}$$

state-transition costs for loads are considered in the same way as for generators which requires those costs to be minimized, leading to the above form.

It is assumed for this class of units that the power variable $g(t)$ $(y(t))$ is continuous, and the objective function P^g (P^l) is also continuous and convex (concave). That is, the constraint set $\mathcal{S}^g(t)$ $(\mathcal{S}^l(t))$ should be described as a continuous interval. The transition cost will typically have the following form,

$$S^g(x^g(t), u^g(t)) = \begin{cases} c_1(t), & \text{if } (x^g(t), u^g(t)) = (0, 0) \\ c_2(t), & \text{if } (x^g(t), u^g(t)) = (0, 1) \\ c_3(t), & \text{if } (x^g(t), u^g(t)) = (1, 0) \\ c_4(t), & \text{if } (x^g(t), u^g(t)) = (1, 1); \end{cases} \tag{2.11}$$

the constants $c_i(t)$ should be non-negative and can in general be time varying. Examples include standard electrical appliances such as lighting systems or a microwave oven in an off-grid home.

Example 2 (Up/down time accumulation cost) Another example considers a unit with $n = 2$ states with a cost (utility) that is a function of the power level, transition states, and the cumulative time that the unit is ‘on’ or ‘off’. For example, when a dish washer or oven is scheduled for use, the utility obtained by the user could potentially be degraded if the load must be turned off during its operation. The dynamics can be written as

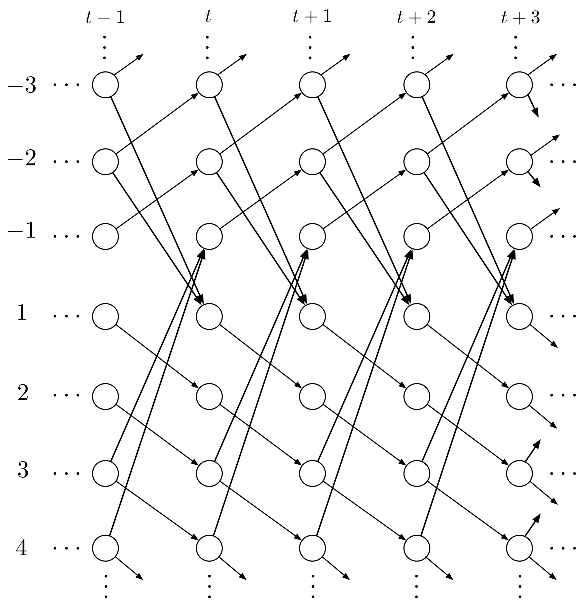
$$x^g(t+1) = \begin{cases} x^g(t) + u^g(t), & \text{if } x^g(t)u^g(t) > 0 \\ u^g(t), & \text{if } x^g(t)u^g(t) < 0, \end{cases} \tag{2.12}$$

with $x^g(t) \in \mathcal{X} = \{-T, \ldots, -2, -1, 1, 2, \ldots, T\}$ and $u^g(t) \in \mathcal{U} = \{-1, 1\}$, for all $t \in \mathcal{T}$. Here we take positive values of $x^g(t)$ to denote the number of time steps the unit is ‘on’, and negative values for the number of time steps the unit is ‘off’. Therefore, the ‘on’ state corresponds to $x(t) > 0$, and ‘off’ to $x(t) < 0$. This is illustrated with a state transition diagram shown in Fig. 2 (the diagram also includes minimum on and off time constraints, discussed below).

The cost function can be decomposed in a similar manner as in (2.9). As in the previous example, this class of unit requires that the constraint variables $\mathcal{S}^g(t)$ and $\mathcal{S}^l(t)$ are continuous intervals. The cost function for such a setting would most naturally be described by a piece-wise continuous (and convex/concave) function. In this way, the time that the unit is in the ‘on’ or ‘off’ state would constitute break-points for the objective function.

[4]In this way, the transition cost reflects a change in state at time t to time $(t+1)$; the transition cost could similarly be defined to reflect the change from time $(t-1)$ to t, representing a slightly different interpretation for the total cost at time t.

Fig. 2 State transition diagram showing accumulation of time spent in 'on' or 'off' states with minimum on ($\tau_{up} = 3$) and off time ($\tau_{down} = 2$) constraints



Example 3 (Minimum up/down time) Certain generating units and loads may also require a minimum up or down time constraint for their operation. This constraint introduces a non-linearity into the dynamics given in (2.12) forcing $u^g(t) = 1$ so long as $x^g(t) \leq \tau_{up}$, or $u^g(t) = -1$ so long as $x^g(t) \geq -\tau_{down}$; here τ_{up} and τ_{down} are positive integers representing the minimum time the unit must remain on or off. Figure 2 shows the state transition diagram for such a unit, with $\tau_{up} = 3$ and $\tau_{down} = 2$. The constraints are represented by not allowing transitions between on and off states unless the unit is in the appropriate state.

Remark 3 The three examples presented above illustrate the type of problems that can be addressed by the class *CCn*. It is worth emphasizing, however, that problems with convex and continuous objectives fall under a very general and broad class of optimization problems, all of which admit efficient solution algorithms. Of interest to this work are the class of problems known as *monotropic programming* covering a wide variety of constrained optimization problems; see, for example, [46].

2.1.3 Discrete objective, n states (Dn)

Another important class of units are those that have in addition to discrete operating states, discrete levels of power generation or consumption. Perhaps the simplest example could be the lighting system in a vehicle; when a light is on it requires a fixed amount of power based on its rating. An example of a generating unit with a similar description could be a battery or super-capacitor that can only output power at a fixed rate. In both examples, the abstract constraint set $\mathcal{S}^g(t)$ ($\mathcal{S}^l(t)$) for the power variable $g(t)$ ($y(t)$) would contain discrete elements. Consequently, the decomposition of the

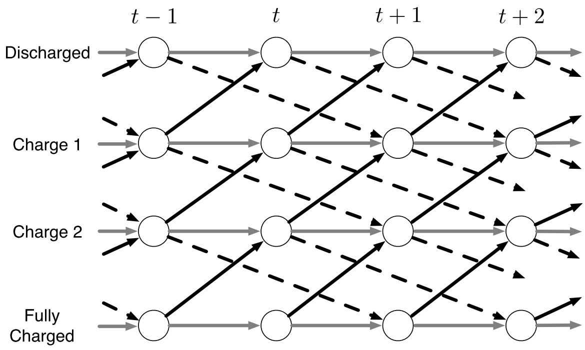


Fig. 3 State transition diagram for a rechargeable battery

cost and utility functions will not posses the convexity and continuity properties of the previous classes of units.

Example 4 (Charging and discharging units) As an illustrative example, we consider a unit that can behave as either a generating unit or a load depending on its state. Rechargeable batteries and super-capacitors fall under this characterization. Figure 3 shows the state transition diagram for a rechargeable battery that has four discrete states corresponding to being discharged, at $1/3$ charge, $2/3$ charge, and fully charged. Note that in this example the charge rate is different than the discharge rate; it requires two time steps to charge the unit to the next state (the dashed line in Fig. 3), whereas it is discharged in one time step (the solid black line in Fig. 3). Furthermore, the unit is also able to 'hold' its charge, as indicated by the grey arrows in Fig. 3. The dynamic description of this model must keep track of how long the unit has been charging or discharging. Furthermore, we note that once the unit has committed to charging, it will not be available to discharge (or hold) until it is at the next charge state. The dynamics for a generalized version can be written as

$$x(t+1) = \begin{cases} \min(x(t)+u(t), T_c), & \text{if } x(t)u(t) > 0 \\ \max(x(t)+\frac{r_c}{r_d}u(t), 0), & \text{if } x(t)u(t) < 0 \\ x(t), & \text{if } u(t) = 0, \end{cases} \tag{2.13}$$

where at each time step, $x(t) \in \{r_c, 2r_c, 3r_c, \dots, T_c\}$. Here, T_c (assumed to be an integer) is the time required to charge the unit from a discharged state to the fully charged state, r_c is the charge rate, and r_d is the discharge rate. Note that this description requires T_c to be an integer multiple of the charge rate r_c. For the example in Fig. 3, $T_c = 6$, $r_c = 2$, and $r_d = 1$. In this description, the state variable $x(t)$ represents the partial charge level of the unit. Here, $x(t) \in \mathcal{X} = \{0, 1, 2, \dots, T_c\}$ and $u(t) \in \mathcal{U} = \{-1, 0, 1\}$. For n discrete charge states, $x(t) = 0$ corresponds to the fully discharged state, $x(t) = kr_c$ to k/n charge (for $k = 1, 2, \dots, n$), and $x(t) = T_c$ to the fully charged state.

The state transition costs can in general take the same form as (2.11). For this type of unit, the objective function must be considered as a generating cost while it is discharging, and a utility when it is charging. In this way, the cost should be represented as $C(g(t), x^g(t), u^g(t)) = P(g(t), x^g(t), u^g(t)) + S^g(x^g(t), u^g(t))$; although the unit can be considered as a load, we use the notation of the generating unit with the book keeping requirement that when $u(t) = -1$ the unit should be treated as a load.

Another important observation relates to the precise form of the function $P(g(t), x^g(t), u^g(t))$. In general, the utility function and cost function will be different, and the objective can be written as

$$P(g(t), x^g(t), u^g(t)) = \begin{cases} P^g(g(t), x^g(t)), & u(t) = 1 \\ P^l(g(t), x^g(t)), & u(t) = -1 \\ 0, & u(t) = 0. \end{cases} \tag{2.14}$$

We emphasize that as $g(t) \in \mathcal{S}^g(t)$ is a discrete element, the functions $P^g(g(t), x^g(t))$ and $P^l(g(t), x^g(t))$ will be discontinuous and consequently non-convex.

An important abstraction here is the emphasis on a qualitative description of the unit via the state transition diagram as opposed to a dynamic description, as in (2.13). Note that the model derived in (2.13) is a switched non-linear system. It is not hard to construct additional models that have relatively simple qualitative descriptions, but highly complex mathematical models. We will show in the sequel that in most instances a qualitative description is sufficient to solve the problem $\mathcal{P}$.

Another class of units that also falls into this class include those that do not have any explicit cost or utility function. For example, a battery might not have any explicit cost for discharging; there is no fuel consumption or maintenance required. Similarly, the utility of charging a battery might also be difficult to characterize. In this way we are able to set $P(g(t), x^g(t), u^g(t)) = 0$, and the primary objective becomes the scheduling of charging and discharging the battery.

2.1.4 Additional remarks

The above descriptions provide a powerful framework for dealing with a variety units. We also refer the reader to [47] for a detailed exposition of how these methods can also be applied to units with non-linear objectives and constraints, such as *ramping constraints*. It is worth emphasizing, therefore, that the benefit of this modeling framework lies in its ability to handle nonlinear objectives with discrete and state-dependent constraints in a *scalable* fashion. This becomes more important when discussing solution methods for these problems, provided in the sequel.

3 Solution methods

The primary objective is to solve $\mathcal{P}$ in a distributed manner. In the absence of the coupling power balance constraint (2.7), problem $\mathcal{P}$ can be decomposed into subproblems involving only the cost and constraint for each individual unit. However, by relaxing the coupling constraint into the objective, we arrive at a problem formulation that lends itself to a straightforward decomposition. This approach is standard and is described in detail in, for example, [48]; we only provide a brief overview here.

To begin, we introduce the Lagrange multiplier $\lambda \in \mathbb{R}^T$ and define the Lagrangian function as

$$\mathbf{L}(\mathbf{g}, \mathbf{y}, \mathbf{x}^g, \mathbf{x}^l, \mathbf{u}^g, \mathbf{u}^l, \lambda) = J(\mathbf{g}, \mathbf{y}, \mathbf{x}^g, \mathbf{x}^l, \mathbf{u}^g, \mathbf{u}^l) + \sum_{t=1}^{T} \lambda_t \left(\sum_{j=1}^{L} y_j(t) - \sum_{i=1}^{G} g_i(t) \right). \tag{3.1}$$

From the Lagrangian (3.1) we can define the dual function as

$$q(\lambda) = \min_{\mathbf{g}, \mathbf{y}, \mathbf{x}^g, \mathbf{x}^l, \mathbf{u}^g, \mathbf{u}^l} \mathbf{L}(\mathbf{g}, \mathbf{y}, \mathbf{x}^g, \mathbf{x}^l, \mathbf{u}^g, \mathbf{u}^l, \lambda); \tag{3.2}$$

the minimization problem (3.2) is subject to the constraints (2.3)–(2.6). The dual function lends an economic interpretation to the original problem $\mathcal{P}$. The multiplier λ can be considered as a price per unit of power; when the power balance is positive (i.e., there is more demand than available power), the deficit must be purchased at a cost of λ, whereas if the power balance is negative, the excess can be sold off at a rate of λ.

The most critical feature of the dual function (3.2) is it can be naturally decomposed into subproblems corresponding to each unit as,

$$q_i(\lambda) = \min \sum_{t=1}^{T} \left(C_i(g_i(t), x_i^g(t), u_i^g(t)) - \lambda_t g_i(t) \right) \tag{3.3}$$

$$q_j(\lambda) = \min \sum_{t=1}^{T} \left(U_j(y_j(t), x_j^l(t), u_j^l(t)) + \lambda_t y_j(t) \right). \tag{3.4}$$

Thus, for a fixed value of λ, the problems (3.3) and (3.4) can be solved using an appropriate solver.

Remark 4 The above formulation suggests that each unit has associated with it the ability to perform computation. Without any loss of generality, we note that the sub-problems can be solved at any designated "computation node." This node may in general solve more than one sub-problem; that is it is not a requirement that each sub-problem is solved independently.

The dual function (3.2), in turn, is used to describe the dual optimization problem to (2.2), which we term $\mathcal{D}$. Therefore, using the decompositions shown in (3.3) and (3.4), we can express the dual problems for the generating units and loads as

$$\max_{\lambda} q_i(\lambda), \quad i \in \mathcal{G} \tag{3.5}$$

$$\max_{\lambda} q_j(\lambda), \quad j \in \mathcal{L}. \tag{3.6}$$

It is a well-known result from optimization theory that the optimal value of the primal problem $\mathcal{P}$, which we denote as J^*, constitutes an upper bound for the optimal value of the dual problem $\mathcal{D}$, denoted as q^* [45]; that is $q^* \leq J^*$. We will discuss a solution method for solving the dual problem in the sequel, in addition to recovering the primal solution. Before we delve into the solution methods, we first describe how the dual subproblems, as in (3.3) and (3.4), leads to tractable solution methods for a variety of unit definitions.

The solution method we use for solving the dual problem (3.5) relies on sub-gradient methods [48]. The general procedure involves iteratively updating the Lagrange multiplier value λ in such a way as to maximize the dual function. At each iteration and for a fixed value of λ, the subproblems (3.3) and (3.4) must be solved. It is important to emphasize that while the majority of the algorithmic work occurs in parallel via the solution of each subproblem, the sub-gradient methods requires a coordination step to compute a sub-gradient and update the Lagrange multiplier.

First, we note that the t-th component of the sub-gradient of the dual function can be expressed as

$$\nu(\lambda_t) = \left(\sum_{j=1}^{L} y_j(t) - \sum_{i=1}^{G} g_i(t) \right). \tag{3.7}$$

The index t ranges over the entire time horizon, $\mathcal{T}$. The sub-gradient, which provides an ascent direction for the dual problem, is precisely the power balance excess or deficit.

Using the sub-gradient we are able to compute an ascent direction for the Lagrange multiplier. Introducing the index k to denote the iteration step, we can compute the update as[5]

$$\lambda^{k+1} = \lambda^k + \alpha^k \nu(\lambda^k). \tag{3.8}$$

The parameter α^k represents the step-size for the update at each iteration. The choice of the step-size will have implications for both the absolute convergence properties of the algorithm and the speed of convergence. The precise choice is highly dependent on the particular application, and selection of this parameter should be approached as a variable requiring iterative tuning. For this work, we consider a step size that is *non-summable* and *diminishing*; that is for each iteration step k,

$$\alpha^k \geq 0, \quad \lim_{k \to \infty} \alpha^k = 0, \quad \text{and} \quad \sum_{k=1}^{\infty} \alpha^k = \infty.$$

The algorithm also requires a stopping criteria which will have implications for the running speed as well as how “good” the solution is. There are some theoretical

[5] For an equality constraint, as in (2.7), the multiplier is unconstrained. If, however, the power balance constraint was written as an inequality constraint, the multiplier update (3.8) would have to be projected onto the positive orthant.

justifications for choosing stopping conditions (and step-size rules) when an optimal solution to the problem is known *a priori* [48]. However, in the absence of this knowledge a more *ad hoc* stopping criteria must be used. For example, in systems where the running time of the algorithm is critical, it may be advantageous to use a fixed number of iterations. While this method may lead to less than desirable solutions, feasibility can still be guaranteed. Another possibility is to use a thresholding technique, whereas if $\|\lambda^{k+1} - \lambda^k\| < \epsilon$, the algorithm stops.

As a final remark, we emphasize that this algorithm provides only *asymptotic* guarantees on convergence. In real implementations, use of a stopping criteria as suggested above will lead to a sub-optimal solution. Furthermore, we note that the ability to reconstruct the global optimum from this algorithm will greatly depend on the specific problem structure and the primal recovery step, that we discuss in Sect. 3.2.

3.1 Unit subproblems

As described above, at each iteration step of the sub-gradient algorithm, the subproblems (3.3) and (3.4) must be solved. The classification of units given in Sect. 2.1 will lead to insight on appropriate solution methods that we present here. It is important to emphasize that one advantage of this method is the flexibility inherited for solving each subproblem. Indeed, as shown in the following, certain sub-problems may admit specialized algorithms or even analytical solutions parameterized by the dual variables. The appropriate choice of the solution method will depend on the particular problem instance, and we emphasize in this discussion the flexibility obtained by considering a shortest-path formulation for many of the problem classes.

3.1.1 Convex and continuous objective, 1 state (CC1)

The sub-problem associated with this class must only consider the power generation or demand level for each unit. The sub-problems (3.3) and (3.4) reduce to

$$q_i(\lambda) = \min_{g_i(t) \in \mathcal{S}_i^g(t)} \sum_{t=1}^{T} \big(C_i(g_i(t)) - \lambda_t g_i(t) \big), \quad \forall i \in \mathcal{G} \tag{3.9}$$

$$q_j(\lambda) = \min_{y_j(t) \in \mathcal{S}_j^l(t)} \sum_{t=1}^{T} \big(U_j(y_j(t)) + \lambda_t y_j(t) \big), \quad \forall j \in \mathcal{L}. \tag{3.10}$$

Recall that for this class of units, the objective functions $C_i(g_i(t))$ and $U_j(y_j(t))$ are continuous and convex functions. Consequently, the sub-problems (3.9)–(3.10) are also convex (concave) lending to efficient algorithms for finding their solutions. In the most general form, these functions fall under the class of convex programming and the specific form of the objectives will dictate the appropriate solution method (e.g., linear program or quadratic program). Furthermore, for certain classes of objective functions analytic solutions can be computed *a priori* parameterized by the multiplier value λ. In this way, the solution to these sub-problems can be efficiently computed with minimal computational overhead.

3.1.2 Convex and continuous objective, n states (CCn)

This class of units is the most closely related to units described in traditional ED and UTC literature [22–26]. The combination of the scheduling problem with the commitment level of each unit is most readily solved using techniques from dynamic programming [49]. As in the case with the class *CC1*, the introduction of the term $-\lambda_t g_i(t)$ and $\lambda_t y_j(t)$ does not cause the objective function to lose its convexity properties.

3.1.3 Discrete objective, n states (Dn)

Recall that the class *Dn* contains discrete decision variables with a discontinuous and non-convex objective function description. This poses a computational challenge for solving the sub-problems (3.3) and (3.4). Fortunately, the qualitative description of the unit's operation via the state-transition diagram leads to insight on how to solve the sub-problems.

To illustrate the procedure, we return to the rechargeable battery example of Sect. 2.1.3. We will assume that there is a constant operating cost when the super-capacitor is charging, and there is a constant utility for when it is discharging. The cost functions specified in (2.14) have the form

$$P^g(g(t), x^g(t)) = \gamma_d \quad \text{and} \quad P^l(g(t), x^g(t)) = \gamma_c. \tag{3.11}$$

The key observation here is the discrete nature of the power variable $g(t)$ can be treated in the same way as the discrete state and decision variables. With such a perspective, tools from dynamic programming can be used. More intuitive, however, is the use of shortest path algorithms to solve the sub-problems [49]. Given the state-transition graph of Fig. 3 we must simply assign costs to each edge and then use an appropriate algorithm such as Dijkstra's or the Bellman-Ford algorithm, to solve the problem. What is important to note is the costs of certain edges will be time-dependent. For example, assume the constraint set for the power variable $g(t)$ is $\mathcal{S}^g = \{-p, 0, p\}$ for all $t \in \mathcal{T}$; this corresponds to a constant charge and discharge rate when the battery is in the appropriate state. The decision to transition from a fully discharged state at time t to the next charge state (which is reached at time $t+2$), will incur a transition cost of $S^g(0, 1)$, obtain a utility of $P^l(-p, 0) = \gamma_c$, and pay a "phantom" price of $\lambda_t p$. The dashed line edge from Fig. 3 from time t to $t+2$ therefore gets assigned a cost of $(S^g(0, 1) - \gamma_c + \lambda_t p)$; note that the utility gained is *subtracted* from the total edge cost, as we would like to minimize the objective.

The shortest path approach to solving these subproblems also lends itself to a more transparent understanding of how the sub-gradient algorithm is working. At each iteration, the values of the multipliers get updated in such a way to solve the dual problem $\mathcal{D}$. Consequently, the shortest path solution is expected to change at each iteration as a result of the update equation (3.8), and this can be monitored to understand the impact of each unit on the aggregate solution.

Finally, we must "condition" the state transition diagram to accommodate the shortest path algorithm. A simple approach is to add a 'Start' node (labeled S) and a 'Terminal' node (labeled T) to the graph, as shown in Fig. 4. Edges connecting these

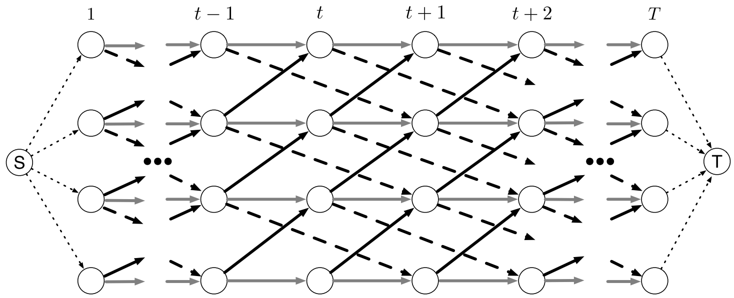


Fig. 4 Shortest path graph for a rechargeable battery

augmented nodes can be used to specify initial and terminal operating states. For example, if the initial state is fully charged, then there should only be one edge leaving the start node and connecting to the state corresponding to being fully charged. Similarly, if initial and final state conditions are to be determined via optimization, all edges can be included, possibly assigning additional costs reflecting those decisions.

As discussed in (3.3), the class *Dn* can also include units with no explicit cost or utility function associated with them. For example, in the supercapacitor example, we might have $\gamma_d = \gamma_c = 0$. Note that this change does not affect the general procedure, and we note that the Lagrange multiplier still introduces a "phantom" price for the consumption or production of power at a specified level. This most clearly illustrates the hidden costs of, for example, charging or discharging a battery or supercapacitor. While no explicit cost is provided, there is an implicit cost incurred by the rest of the power system in order to meet the state requirement of the unit.

3.2 Primal recovery

An important final step in the sub-gradient algorithm described above is the recovery of the primal solution to the problem $\mathcal{P}$ from the solution of the dual problem $\mathcal{D}$. Indeed, for special classes of optimization problems, such as those that are strictly convex, the recovery of the primal solution from the dual is a straight-forward procedure. In such cases, a convex combination of primal variables from the subproblems are used to generate the primal solution [48]. For example, consider a scenario where all generating units and loads are of the class *CC1*. At each iteration step in the sub-gradient algorithm, the sub-problems (3.3) and (3.4) must be solved. It can be shown that in the limit, the primal solution can be obtained from a convex combination of the solution of from each iteration in the algorithm. To better illustrate this point, consider a generating unit with optimal commitment level $g^*(t)$, and denote the optimal solution of the k-th iteration in the sub-gradient algorithm as $g^{k*}(t)$. Then the primal solution can be expressed as [48]

$$g^*(t) = \lim_{k\to\infty} \frac{\sum_{r=1}^{k} \alpha^r g^{r*}(t)}{\sum_{r=1}^{k} \alpha^r}. \tag{3.12}$$

Note that when the variables are continuous, that is when the constraint sets $\mathcal{S}^g(t)$ or $\mathcal{S}^l(t)$ represent continuous intervals or box constraints, then the convex combination will be guaranteed to be feasible [45].

However, in the general set-up developed here, we do not have such properties. Clearly, the method used in (3.12) can not work when the primal variables are discrete; a convex combination of discrete points will not in general correspond to a feasible solution. In fact, even for linear programming examples, customized solution methods must be employed for primal recovery, such as bundle methods [50]. Furthermore, existing literature on primal recovery techniques for mixed integer nonlinear programs is scarce. As a result, we propose a heuristic method for recovering a feasible primal solution from the dual problem. Note that with this heuristic we are no longer able to make guarantees about the optimality of these solution, but rather emphasize that feasibility is ensured.

The first important observation is that each subproblem guarantees that the solutions will be feasible in the absence of the power balance constraint (2.7). The main challenge for primal recovery, therefore, is that the power balance (2.7) is satisfied. In this direction, we propose to use a combination of the primal recovery technique shown in (3.12) with a load and source shedding heuristic.

As the classification of units suggest, the problem $\mathcal{P}$ will contain a combination of both continuous and discrete variables representing the power generation and consumption of units, along with a set of discrete variables representing the state and control variables for each unit. The primal recovery heuristic is described below. We denote by k^* the last iteration count of the algorithm.

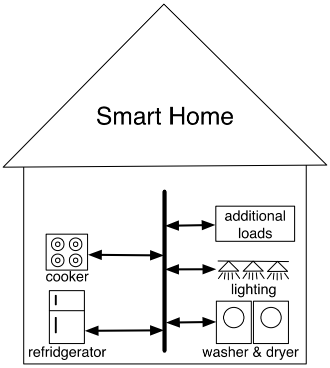
Algorithm 1 Primal Recovery Heuristic

1. For each unit belonging to the class *CC1* and *CCn*, construct the primal power level variables $g_i(t)$ and $y_j(t)$ using (3.12). The state and control variables for these units will be taken from the solution of the sub-problems (3.3) and (3.4) at the last iteration step k^*.
2. For each unit belonging to the class *Dn*, use the solution for the power level variable and the state and control variables from the last iteration step; that is the values $g_i(t)$, $x_i^g(t)$, $u_i^g(t)$, $y_j(t)$, $x_j^l(t)$, and $u_j^l(t)$ at iteration k^*.
3. Check the power balance constraint (2.7).
4. If the power balance is satisfied, use that solution. Otherwise, begin shedding loads or increasing power generation according to a predetermined priority list.

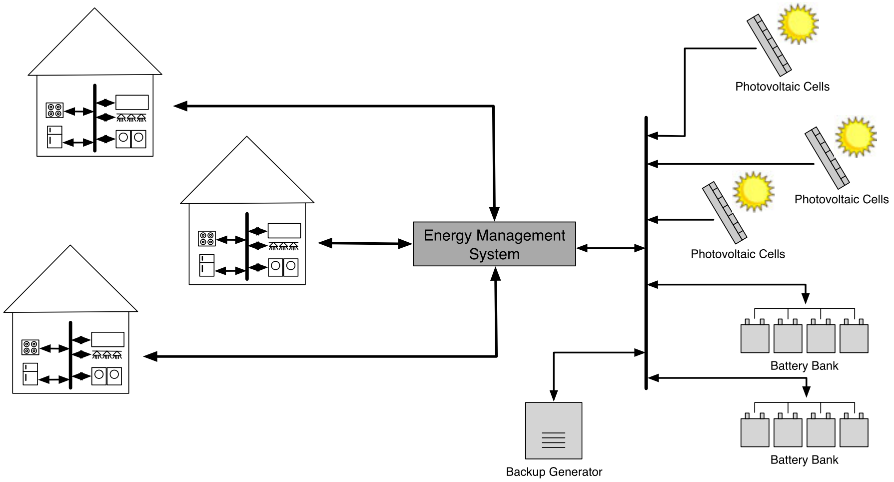
The last step in the procedure deserves some elucidation. One approach is to decide if the primal recovery should operate as a *utility maximization priority*, *generation cost priority*, or *aggregate cost priority* (e.g., via the introduction of and appropriate choice of a weighting constant κ, discussed in Remark 1). For example, if load utility has a greater overall importance to the operation of the power system, then when the power balance is not met, generation of power should be increased when available. Similarly, if the generation cost is higher priority, than loads should be shed or their demand level lowered when possible. In either situation, we find that this procedure now relates to techniques related to load shedding heuristics and priority assignments [36, 37, 39–44]. We would like to note that in our experience, the power balance constraint is satisfied without the need for any additional shedding.

4 Simulation example: an off-grid solar powered community

As illustrated in Fig. 5, we provide here an example of an off-grid community including three smart homes equipped with multiple power generation sources and schedulable electric appliances to demonstrate how this algorithm can efficiently schedule their operations. The primary power sources used in this example are three sets of solar panels for medium sized homes with an array size of 3.5 kW and approximate solar isolation of 5 hours during spring time for each set [51]. Considering the differ-



(a) A smart home.



(b) An off-grid community.

Fig. 5 An off-grid solar powered community of smart homes

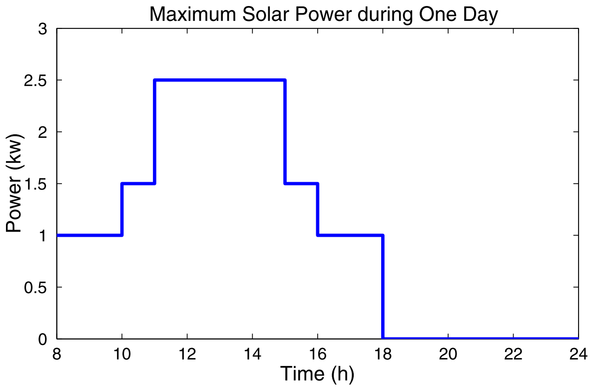


Fig. 6 Maximum solar power during one day

ence of the isolation strength during one day, we allocate the maximum power output from 8 a.m. to 24 p.m. for one set of solar source in Fig. 6. The power generation cost is due to the capital investment of solar panels and maintenance fees. The solar system in [51] for a medium sized home has a list price of $11,700, with an expected monthly output of 529 kWh in California area, and free life-time maintenance. Considering the 30% federal tax return and $1000–$2000 state rebate, the solar system with 30 year usage has electricity cost around $0.04 per kWh. The community is also equipped with two sets of battery banks capable of supplying a maximum of 6 kW power for 6 continuous hours. Considering the battery life cycle of 3300 and investment fee of $2870 for one set, we assign the cost for power provided from the battery at $0.08 per kWh if the battery set is 60% charged for each cycle on average. The battery sets are assumed to be half charged at the beginning time in this simulation example. This unit falls under the class *Dn*. Under some emergency situations, e.g., continuous rainy days, a backup generator is necessary. Here, the backup generator is a small size diesel generator with fixed 8 kW power output which can only be supplied for 30 minutes before refueling. During the refueling state, the diesel generator must be switched to an 'off' state for another 30 minutes before it can be available again; this is modeled after *Example 3* describing units with a minimum up and down time constraint. Considering the current market price of diesel at $4 per gallon, the electricity cost from the backup generator has a much higher cost at $0.50 per kWh.

In this example, we demonstrate how appropriate modeling of the load utility functions and constraints are equivalent to assigning a priority for each load. The advantage for using a modeling framework is that the notion of priority is now captured via the optimization, rather than through a heuristic. In this direction, we model the disutility[6] of each load as an non-decreasing function of the time the load is not 'on.' That is, if a load is turned 'off' in order to satisfy the power balance, the corresponding disutility will increase. For all of the appliances included in this example, we assume the disutility is constant over fixed intervals, and the accumulative disutility value increases as the time in the 'off' state is longer; this is modeled after *Example 2* describing units with up and down time accumulation costs. Each load, therefore, will

[6]Note that for the remainder of this example we refer to 'disutility' as opposed to 'utility.' This is without loss of generality and translates only to a sign change for the load segment of (2.1).

be modeled under the class Dn. The constraints for each load are also modeled as equality constraints; when a load is in the 'on' state it will demand a fixed level of power, and when it is 'off' it requires no power. The power demand constraint and disutility values of the all load units from three homes are listed in Table 1, where P denotes the power demand from each load referring to the example of [10], c represents the disutility cost for being in the 'off' state during the fixed interval, $t0$ is the operation request starting time and Δt is the requested duration time. The subscript 1, 2 and 3 indicates the index of the home. The disutility cost is assigned according to the priority of the appliances in the daily life. Some units, i.e., the refrigerator and lighting system, will bring more inconvenience if purposely paused during a request time. For these units, we assign a higher disutility cost than the others to guarantee their normal operation provided there is enough power available. Other units, i.e., the dish washer, which is not urgent to be scheduled right upon request, is assigned lower disutility cost.

From the scenario description in Table 1, the overall daily energy demand request from the small community is 61.73 kWh which is near the solar daily output capability of 52.5 kWh. Considering that some appliances, i.e., the washing machine and spin drier, may not be used everyday, the proposed generation system of a photovoltaic array, battery, and backup generator can generally meet the demand requirements of this community with three medium sized homes. However, during certain peak request periods, i.e., in the mornings and evenings, conflicts can arise when there are more load requests than available power; there will not be enough generating capability from the solar source alone to supply the loads, especially when the maximum solar power output is small due to isolation strength during those time intervals. It is expected that the energy management system can efficiently schedule the operation to alleviate the demands during peak period and at the same time avoid shutting down any unit too long leading to increased disutility.

For this example we also implement a 'receding horizon' type of approach to simulate the user-driven nature of the loads. We consider a moving time horizon for the optimizer from the current request time to the end of the day (24 p.m.) with each time step corresponding to 30 minute intervals. At the initial time, the algorithm determines the optimal schedule and allocation for each source and load. Here we point out that the optimizer can not anticipate that, for example, the dish washer from home 1 will turn on at 9 a.m. As a result, the optimizer uses the current request level for each load and assumes it will be active until the end of its duration time. When a new load request is initiated, the optimizer must recompute the schedule and allocation for all units using the new state of the system. Otherwise, the generated optimal schedule will not change until the end of the day.

The Lagrange multipliers in the sub-gradient algorithm is initialized to zero and we use $\alpha^k(t) = \frac{0.1}{\sqrt{k}\,|\nu(\lambda^k)|}$ for the step-size α^k in (3.8) to ensure the speed of convergence is in a controlled scope. The optimal schedule for this scenario is illustrated in Fig. 7 and Table 2 for the source's power outputs and load's operation histories, i.e., starting time ts, ending time te and delayed time td, respectively.

From Fig. 7, we observe that solar sources, as the cheapest power supply unit in this system, will provide as much power as required by the loads if the request is under its maximum power output limit. When the request is above solar source's upper

Table 1 Constraint, disutility and operation scenario for loads

Unit name	P (kW)	c_1 ($)	$t0_1$ (h)	Δt_1 (h)	c_2 ($)	$t0_2$ (h)	Δt_2 (h)	c_3 ($)	$t0_3$ (h)	Δt_3 (h)
Cooker hob	3	0.1	8	0.5	0.09	8	0.5	0.08	8.5	0.5
Microwave	1.7	0.1	8	0.5	0.09	n/a	n/a	0.08	n/a	n/a
Dish washer	1	0.002	9	1	0.0018	9.5	1	0.0016	11	1
Clothes washer	1	0.02	9	1.5	0.018	n/a	n/a	0.016	8	1
Vacuum robot	1.2	0.002	9	0.5	0.0018	13	1	0.0016	8	0.5
Dryer	2.5	0.02	14	1	0.018	n/a	n/a	0.016	12	1
Cooker oven	5	0.2	18	0.5	0.18	17.5	0.5	0.16	18	0.5
Lighting	0.84	50	18	6	50	18	6	50	18	6
Desktop	0.3	30	18	3	30	n/a	n/a	20	n/a	n/a
Jacuzzi pump	1.8	0.05	19	1	0.045	19	1	0.04	n/a	n/a
Laptop	0.1	20	20	3	20	n/a	n/a	20	n/a	n/a
Refrigerator	0.3	10	0	24	10	0	24	10	0	24

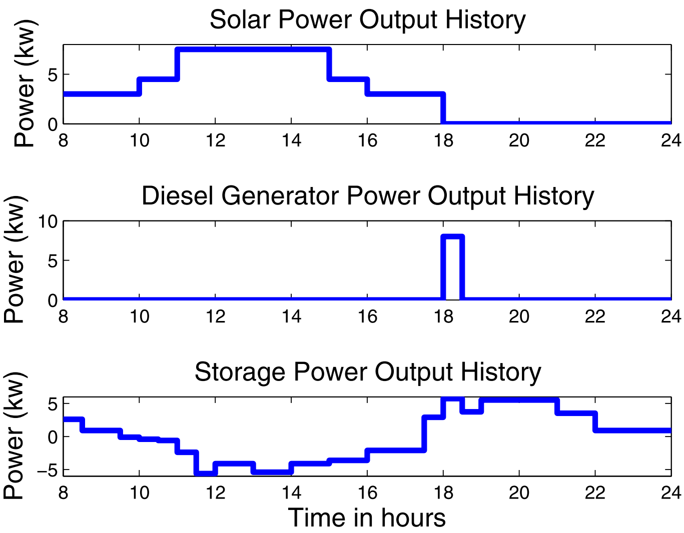


Fig. 7 Sources schedule and commitment levels

Table 2 Optimal schedule of loads operation

Unit name	ts_1 (h)	te_1 (h)	td_1 (h)	ts_2 (h)	te_2 (h)	td_2 (h)	ts_3 (h)	te_3 (h)	td_3 (h)
Cooker hob	8	8.5	0	8.5	9	0.5	9	9.5	0.5
Microwave	8	8.5	0	n/a	n/a	n/a	n/a	n/a	n/a
Dish washer	10.5	11.5	1.5	10.5	11.5	1	11	12	0
Clothes washer	9.5	11	0.5	n/a	n/a	n/a	9.5	10.5	1.5
Vacuum robot	10	10.5	1	13	14	0	11	11.5	3
Spin dryer	14	15	0	n/a	n/a	n/a	12	13	0
Cooker oven	18	18.5	0	17.5	18	0	18	18.5	0
Lighting	18	24	0	18	24	0	18	24	0
Desktop	18	20	0	n/a	n/a	n/a	n/a	n/a	n/a
Jacuzzi pump	19	20	0	20	21	1	n/a	n/a	n/a
Laptop	20	23	0	n/a	n/a	n/a	n/a	n/a	n/a
Refrigerator	8	24	0	8	24	0	8	24	0

bound, the battery, as the second cheapest supply unit, will begin to output power or some units are temporarily shut down until more power is available. For example, during the morning hours 8 a.m. to 10 a.m., when the solar source maximum power output is low, the battery will supply the extra power required for some important appliances, e.g. the cooker hob. However, when the solar power output is increased during noon and afternoon, the battery stores the extra generated energy after consumption. At the 18 p.m. mark, the aggregate requested power from the loads exceed the battery supply limit, initiating the backup generator to supply power. However, since the backup generator can only supply power for 30 minute intervals, the opti-

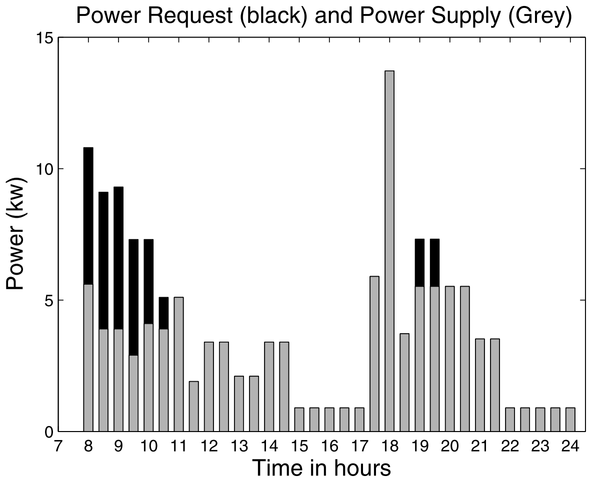


Fig. 8 Aggregate demand levels and delivered power. *Black colored bars* indicate intervals when the load request could not be met and appropriate scheduling of the sources and loads are required

mizer must schedule the demand level of the loads to accommodate for the switching level of available power. From the load operation history of Table 2, at the beginning, appliances having low disutility cost are shut down temporarily to let the more important appliances operate first with power supply coming from the solar source and the battery, which avoids using the backup generator allowing both of them to operate simultaneously. This behavior is due to the net benefit considered by both the cost of running the backup generator and the utility lost for turning off some appliances temporarily. The clothes washer, vacuum robot and dish washer, with lower assigned disutility values, are purposely shut down until the solar power output level meets their requests. The Jacuzzi Pump request from home 2 at 19 p.m. is paused for 1 hour to avoid using the backup generator again.

In order to better illustrate the mediation function of the optimal algorithm among all the units in the system, Fig. 8 shows how the aggregate available power is reduced during peak load request time periods. In this plot, the grey bar is the power supply and the black bar is the power request. When power supply can meet the power request, the grey bar will cover the black bar. Otherwise, the grey bar is less than the black bar. Although the calculation time highly depends on the complexity of the shortest path models, the number of units, and the time horizon of the planning window, in this example with the prescribed scenario and parameters, the calculation time of every new schedule running in Matlab is estimated to be less than 3 seconds when all of the units shortest path solution are solved in sequence using a Lenovo X201 laptop with intel i5 CPU and 4GB RAM. Therefore, if the shortest path solution for all the units are calculated parallel, we can expect to obtain the optimal solution for each time interval in less than 1 second.

The performance of our algorithm is also compared with a direct scheme which has no schedule and will supply the requested power right at the proposed time in Fig. 9; this solution will use the back-up generator regardless of cost. With the optimal schedule, the overall cost is greatly reduced from \$12.86 to \$8.42. With the optimal schedule, it will save approximate \$133.2 in one month and \$1598.4 in one year, which is not a trivial amount.

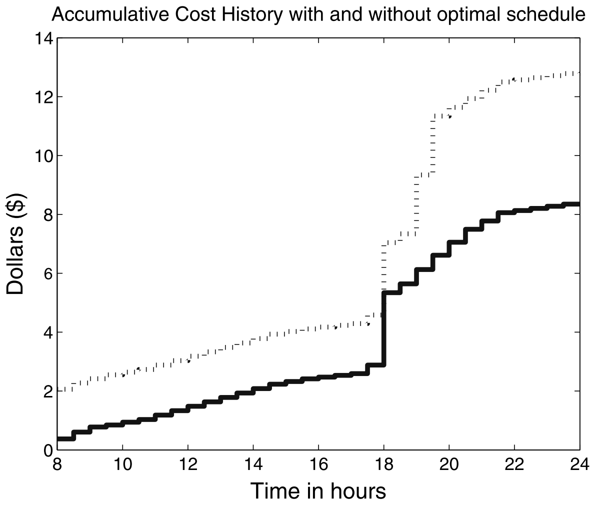


Fig. 9 Cumulative cost history comparing sub-gradient algorithm and a direct scheme without optimal schedule

5 Concluding remarks

This work described an optimization framework for an energy management system of an off-grid power system. The distinguishing feature of this work is the inclusion of load scheduling and utility maximization in the context of a self-contained power system. This work also described how qualitative descriptions of a wide class of generating units and loads can be used to derive alternative representations that lead to tractable solution methods via the shortest path algorithm.

The advent of "smart" power systems of all scales must lead to new perspectives on the role of loads in these systems. While load shedding and other heuristic techniques offer simple and fast methods for guaranteeing the power balance of the system, they are lacking from the perspective of optimizing the operation of the entire power system. Self contained off-grid power systems should approach power management with a more holistic approach to ensure the optimal and efficient operation in terms of both costs and utilities. As the methods for off-grids systems mature, they can also be applied to microgrid and smart-grid architectures, which is the subject of future work.

Acknowledgements The authors would like to thank the associate editor and reviewers for the in-depth comments and discussions regarding this work.

References

1. US department of energy, smart grid. http://www.oe.energy.gov/smartgrid.htm
2. European commission on energy, smartgrids. http://www.smartgrids.eu/
3. Ipakchi, A., Albuyeh, F.: Grid of the future. IEEE Power Energy Mag. **7**(2), 52–62 (2009)
4. Fan, J., Borlase, S.: The evolution of distribution. IEEE Power Energy Mag. **7**(2), 63–68 (2009)
5. Farhangi, H.: The path of the smart grid. IEEE Power Energy Mag. **8**(1), 18–28 (2010)
6. Brown, R.E.: Impact of smart grid on distribution system design. In: Power and Energy Society General Meeting—Conversion and Delivery of Electrical Energy in the 21st Century, 2008, pp. 1–4. IEEE Press, New York (2008)

7. Hatziargyriou, N.: Microgrids [guest editorial]. IEEE Power Energy Mag. **6**(3), 26–29 (2008)
8. Katiraei, F., Iravani, R., Hatziargyriou, N., Dimeas, A.: Microgrids management. IEEE Power Energy Mag. **6**(3), 54–65 (2008)
9. Thornley, V., Kemsley, R., Barbier, C., Nicholson, G.: User perception of demand side management. In: SmartGrids for Distribution IET-CIRED, Frankfurt, Germany (2008)
10. Zhang, D., Papageorgiou, L.G., Samsatli, N.J., Shah, N.: Optimal scheduling of smart homes energy consumption with microgrid. In: Energy 2011: The First International Conference on Smart Grids, Green Communications and IT Energy-aware Technologies, no. c, pp. 70–75 (2011)
11. Misak, S., Prokop, L.: Off-grid power systems. In: 9th International Conference on Environment and Electrical Engineering, 2010, pp. 14–17. IEEE Press, New York (2010)
12. Muntean, N., Cornea, O., Petrila, D.: A new conversion and control system for a small off—grid wind turbine. In: 12th International Conference on Optimization of Electrical and Electronic Equipment, 2010, pp. 1167–1173. IEEE Press, New York (2010)
13. Leak, M.H., . Rashid, M.: Feasibility of off-grid residential power. In: CONIELECOMP 2011, 21st International Conference on Electrical Communications and Computers, pp. 14–17. IEEE Press, New York (2011)
14. Chan, C.C.: The state of the art of electric, hybrid, and fuel cell vehicles. Proc. IEEE **95**(4), 704–718 (2007)
15. Koot, M., Kessels, J., DeJager, B., Heemels, W., VandenBosch, P., Steinbuch, M.: Energy management strategies for vehicular electric power systems. IEEE Trans. Veh. Technol. **54**(3), 771–782 (2005)
16. Davey, K., Longoria, R., Shutt, W., Carroll, J., Nagaraj, K., Park, J., Rosenwinkesl, T., Wu, W., Arapostathis, A.: Reconfiguration in shipboard power systems. Am. Control Conf. **80**(6), 064501 (2007)
17. Maldonado, M., Shah, N., Cleek, K., Walia, P., Korba, G.: Power Management and Distribution System for a More-Electric Aircraft (MADMEL)-Program Status. IEEE Press, New York (2004)
18. Cloyd, J.: Status of the United States air force's more electric aircraft initiative. IEEE Aerosp. Electron. Syst. Mag. **13**(4), 17–22 (1998)
19. Luongo, C.A., Masson, P.J., Nam, T., Mavris, D., Kim, H.D., Brown, G.V., Waters, M., Hall, D.: Next generation more-electric aircraft: a potential application for HTS superconductors. IEEE Trans. Appl. Supercond. **19**(3), 1055–1068 (2009)
20. Mohsenian-Rad, A.H., Leon-Garcia, A.: Optimal residential load control with price prediction in real-time electricity pricing environments. IEEE Trans. Smart Grid **1**(2), 120–133 (2010)
21. Ricquebourg, V., Menga, D., Durand, D., Marhic, B., Delahoche, L., Loge, C.: The smart home concept: our immediate future. In: 1st IEEE International Conference on E-Learning in Industrial Electronics, 2006, pp. 23–28. IEEE Press, New York (2006)
22. Bertsekas, D.P., Lauer, G.S., Sandell, N.R.J., Posbergh, T.A.: Optimal short-term scheduling of large-scale power systems. IEEE Trans. Autom. Control **AC-28**(1), 1–11 (1983)
23. Gruhl, J., Schweppe, F., Ruane, M.: Unit commitment scheduling of electric power systems. In: System Engineering for Power: Status and Prospects, Henniker, NH, pp. 116–128 (1972)
24. Muckstadt, J., Koenig, S.: An application of Lagrangian relaxation to scheduling in power generation systems. Oper. Res. **25**(3), 387–403 (1977)
25. Kirchmayer, L.K.: Economic Operation of Power Systems. Wiley, New York (1958)
26. Guan, X., Luh, P., Yan, H., Amalfi, J.: An optimization-based method for unit commitment. Electr. Power Energy Syst. **14**(1), 9–17 (1992)
27. Ongsakul, W., Petcharaks, N.: Unit commitment by enhanced adaptive Lagrangian relaxation. IEEE Trans. Power Syst. **19**(1), 620–628 (2004)
28. Papalexopoulos, A.: Optimization based methods for unit commitment: Lagrangian relaxation versus general mixed integer programming. In: IEEE Power Engineering Society General Meeting (IEEE Cat. No. 03CH37491), 2003, pp. 1095–1100. IEEE Press, New York (2003)
29. Li, T., Shahidehpour, M.: Price-based unit commitment: a case of Lagrangian relaxation versus mixed integer programming. IEEE Trans. Power Syst. **20**(4), 2015–2025 (2005)
30. Joo, J.Y., Ilic, M.D.: A multi-layered adaptive load management (alm) system: Information exchange between market participants for efficient and reliable energy use. In: Transmission and Distribution Conference and Exposition, 2010 IEEE PES, April, pp. 1–7 (2010)
31. Fahrioglu, M., Alvarado, F.: Using utility information to calibrate customer demand management behavior models. In: IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 02CH37309), 2002, vol. 1, p. 26. IEEE Press, New York (2002)
32. Samadi, P., Mohsenian-Rad, A.-H., Schober, R., Wong, V.W.S., Jatskevich, J.: Optimal real-time pricing algorithm based on utility maximization for smart grid. In: First IEEE International Conference on Smart Grid Communications, 2010, pp. 415–420. IEEE Press, New York (2010)

33. Pedrasa, M.A.A., Spooner, T.D., MacGill, I.F.: Coordinated scheduling of residential distributed energy resources to optimize smart home energy services. IEEE Trans. Smart Grid **1**(2), 134–143 (2010)
34. Chiang, M., Low, S.H., Calderbank, A.R., Doyle, J.: Layering as optimization decomposition: a mathematical theory of network architectures. Proc. IEEE **95**(1), 255–312 (2007)
35. Nedic, A., Ozdaglar, A.: Subgradient methods in network resource allocation: Rate analysis. In: 42nd Annual Conference on Information Sciences and Systems, 2008, pp. 1189–1194. IEEE Press, New York (2008)
36. Andrade, L., Tenning, C.: Design of Boeing 777 electric system. IEEE Aerosp. Electron. Syst. Mag. **7**(7), 4–11 (1992)
37. Huneault, M., Galiana, F.D.: A survey of the optimal power flow literature. IEEE Trans. Power Syst. **6**(2), 762–770 (1991)
38. Xu, D., Girgis, A.: Optimal load shedding strategy in power systems with distributed generation. In: IEEE Power Engineering Society Winter Meeting. Conference Proceedings (Cat. No. 01CH37194), no. C, 2001, pp. 788–793. IEEE Press, New York (2001)
39. Bhattacharyya, K., Crow, M.: A fuzzy logic based approach to direct load control. IEEE Trans. Power Syst. **11**(2), 708–714 (1996)
40. Cohen, A., Wang, C.: An optimization method for load management scheduling. IEEE Trans. Power Syst. **3**(2), 612–618 (1988)
41. Ng, K.H., Sheble, G.B.: Direct load control—a profit-based load management using linear programming. IEEE Trans. Power Syst. **13**(2), 688–695 (1998)
42. Ramanathan, B., Vittal, V.: A framework for evaluation of advanced direct load control with minimum disruption. IEEE Trans. Power Syst. **23**(4), 1681–1688 (2008)
43. Widergren, S.E.: Demand or request: Will load behave? In: IEEE Power & Energy Society General Meeting, 2009, pp. 1–5. IEEE Press, New York (2009)
44. Ruiz, N., Cobelo, I.N., Oyarzabal, J.: A direct load control model for virtual power plant management. IEEE Trans. Power Syst. **24**(2), 959–966 (2009)
45. Boyd, S.P., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
46. Rockafellar, R.T.: Network Flows and Monotropic Optimization. Wiley, New York (1984)
47. Frangioni, A.: Solving nonlinear single-unit commitment problems with ramping constraints. Oper. Res. **54**, 775 (2006)
48. Ruszczynski, A.: Nonlinear Optimization. Princeton University Press, Princeton (2006)
49. Bertsekas, D.P.: Dynamic Programming and Optimal Control. Athena Scientific, Belmont (2007)
50. Sherali, H., Choi, G.: Recovery of primal solutions when using subgradient optimization methods to solve Lagrangian duals of linear programs. Oper. Res. Lett. **19**(3), 105–113 (1996)
51. Medium ac off grid solar powered system for your home. http://www.wholesalesolar.com/solarpowersystems/medium-2-ac-home-off-grid-solar-power.html