

Solving Dynamic Programming with Supremum Terms in the Objective and Application to Optimal Battery Scheduling for Electricity Consumers Subject to Demand Charges

Morgan Jones, Matthew M. Peet

Abstract—In this paper, we consider the problem of dynamic programming when supremum terms appear in the objective function. Such terms can represent overhead costs associated with the underlying state variables. Specifically, this form of optimization problem can be used to represent optimal scheduling of batteries such as the Tesla Powerwall for electrical consumers subject to demand charges - a charge based on the maximum rate of electricity consumption. These demand charges reflect the cost to the utility of building and maintaining generating capacity. Unfortunately, we show that dynamic programming problems with supremum terms do not satisfy the principle of optimality. However, we also show that the supremum is a special case of the class of forward separable objective functions. To solve the dynamic programming problem, we propose a general class of optimization problems with forward separable objectives. We then show that for any problem in this class, there exists an augmented-state dynamic programming problem which satisfies the principle of optimality and the solutions to which yield solutions to the original forward separable problem. We further generalize this approach to stochastic dynamic programming problems and apply the results to the problem of optimal battery scheduling with demand charges using a data-based stochastic model for electricity usage and solar generation by the consumer.

I. INTRODUCTION

In 2012, 95,000 new distributed solar PhotoVoltaic (PV) systems were installed nationally, a 36% increase from 2011 and yielding a total of approximately 300,000 installations total [1]. Further, utility-scale PV generating capacity has increased at an even faster rate, with 2012 installations more than doubling that of 2011 [2]. Meanwhile, partially due to the development of energy-efficient appliances and new materials for insulation, US electricity demand has plateaued [3]. As a consequence of these trends, utility companies are faced with the problem that demand *peaks* continue to grow. Specifically, as per the US EIA [4], the ratio of peak demand to average demand has increased dramatically over the last 20 years.

Fundamentally, the problem faced by utilities is that consumers are typically charged based on total electricity consumption, while utility costs are based both on consumption and for building and maintaining the generating capacity necessary to meet peak demand. Recently, several public and private utilities have moved to address this imbalance by charging residential consumers based on the maximum rate (\$ per kW) of consumption - a cost referred to as a *demand charge*. Specifically, in Arizona, both major utilities SRP and APS have mandatory demand charges for residential consumers [5].

For consumers, load is relatively inflexible and hence the most direct approach to minimizing the effect of demand charges is the use of battery storage devices such as the Tesla Powerwall [6], [7], [8]. These devices allow consumers to shift electricity consumption away from periods of peak demand, thereby minimizing the effect of demand charges. In this paper, we specifically focus on battery storage coupled with HVAC and solar generation. This is due to the fact that load from HVAC and electricity from solar generation can be forecast well apriori.

The use of battery storage has been well documented in the literature [9] and in particular, there have been several results on the optimal use of batteries for residential customers [10], [11], [12], [13]. Within this literature, there are relatively few results which include demand charges. Of those which do treat demand charges, we mention [14] which proposes a heuristic form of dynamic programming, and the recent work in [15], wherein the optimization problem is broken down into several agents, and a Lagrangian approach is used to perform the optimization. Furthermore, in [16] a similar energy storage problem is solved using optimized curtailment and load shedding. An L_p approximation of the demand charge was used in combination with multi-objective optimization in [17] and, in addition, the optimal use of building mass for energy storage was considered in [18], wherein a bisection on the demand charges was used. However, we note that none of these approaches resolve the fundamental mathematical problem of dynamic programming with a non-separable cost function and hence are either inaccurate, computationally expensive, or are not guaranteed to converge. Finally, we note that there has been no work to date on optimization of demand charges coupled with stochastic models of solar generation.

In this paper, we formulate the battery storage problem as a dynamic program with an objective function consisting of both integrated time-of-use charges and a supremum term representing the demand charge. Furthermore, we model solar generation as a Gauss-Markov process and minimize the expected value of the objective. The fundamental mathematical challenge with dynamic programming problems of this form is that, as shown in Section II, problems which include supremum terms in the objective do not satisfy the *principle of optimality* and thus recursive solution of the Bellman equation does not yield an optimal policy.

Dynamic programming for problems which do not satisfy the principle of optimality has received little attention and

there are few results in the literature in which this problem has been addressed. The only generalized approach to the problem seems to be that taken in [19] which considered the use of multi-objective optimization in the case where the objective function is “backward separable”. Although the supremum term is not backward separable, an L_p approximation of the supremum is backward separable and this approach was applied in [17] to the problem of battery storage. Although not directly addressed in [19], our approach is inspired by this result and is based on the observation that while the supremum is not backward separable, it is “forward separable”.

To solve forward separable optimization problems, we propose in this paper a rigorous approach to a class of generalized dynamic programming problems which are formally defined in Section II. For this class of problems, we propose a precise definition of the *principle of optimality* and show that if this definition holds, then the Bellman equation can be used to define an optimal policy. Next, we propose a class of forward separable optimization problems and show that dynamic programming with integral and supremum terms is an element of this class. We then show that the principle of optimality fails for certain problems in this class. In Section III, we show that for any forward separable dynamic programming problem, there exists a separable augmented-state dynamic programming problem for which the principle of optimality holds and from which solutions to the original forward separable problem can be recovered. In Section IV, we apply these methods to the battery scheduling problem for a given load and solar generation schedule. In Section VI, we show that the augmented dynamic programming problem can also be used to solve stochastic dynamic programming problems with forward separable objectives and apply this approach to the battery scheduling problem using a Gauss-Markov model of solar generation extracted from data provided by local utility SRP.

II. BACKGROUND: GENERALIZED DYNAMIC PROGRAMMING

In this paper, we consider a generalized class of dynamic programming problems. Specifically, we define a generalized dynamic programming problem as an indexed sequence of optimization problems $G(t_0, x_0)$, defined by an indexed sequence of objective functions $J_{t_0, x_0} : \mathbb{R}^{m \times (T-t_0)} \times \mathbb{R}^{n \times (T-t_0+1)}$ where we say that $\mathbf{u}^* \in \mathbb{R}^{m \times (T-t_0)}$ and $\mathbf{x}^* \in \mathbb{R}^{n \times (T-t_0+1)}$ solve $G(t_0, x_0)$ if

$$(\mathbf{u}^*, \mathbf{x}^*) = \arg \min_{\mathbf{u}, \mathbf{x}} J_{t_0, x_0}(\mathbf{u}, \mathbf{x}) \quad (1)$$

subject to: $x(t+1) = f[x(t), u(t), t]$, given $x(t_0) = x_0$

$x(t) \in X \subset \mathbb{R}^n$ for $t = t_0 + 1, \dots, T$

$u(t) \in U \subset \mathbb{R}^m$ for $t = t_0, \dots, T-1$

Where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{N} \rightarrow \mathbb{R}^n$, $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ for all t . We denote $J_{t_0, x_0}^* = J_{t_0, x_0}(\mathbf{u}^*, \mathbf{x}^*)$.

Definition 1: We say the sequence of controls $\mathbf{u} = (u(t_0), \dots, u(T-1)) \in \mathbb{R}^{m \times (T-t_0)}$ is feasible if $u(t) \in U$ for $t = t_0, \dots, T-1$ and if $x(t+1) = f[x(t), u(t), t]$ and

$x(t_0) = x_0$, then $x(t) \in X$ for all t . For a given x , we denote by $\Gamma_{t,x}$, the set $u \in U$ such that $f[x, u, t] \in X$. In this paper we only consider problems where $\Gamma_{t,x}$ is nonempty for all x and t .

Note that for this class of optimization problems, feasibility is inherited. That is, if $\mathbf{u} = (u(t), \dots, u(T-1))$ and $\mathbf{x} = (x(t), \dots, x(T))$ are feasible for $G(t, x(t))$ and $\mathbf{v} = (v(s), \dots, v(T-1))$ and $\mathbf{h} = (h(s), \dots, h(T))$ are feasible for $G(s, x(s))$ where $s > t$, then $\mathbf{w} = (u(t), \dots, u(s-1), v(s), \dots, v(T-1))$ and $\mathbf{z} = (x(t), \dots, x(s-1), h(s), \dots, h(T))$ are feasible for $G(t, x(t))$.

In certain cases, indexed optimization problems of the Form of $G(t_0, x_0)$ can be solved using an optimal policy.

Definition 2: A policy is any map from the present state and time to a feasible input $(x, t) \mapsto u(t) \in \Gamma_{x,t}$, as $u(t) = \pi(x, t)$. We say that π^* is an *optimal policy* for Problem (1) if

$$\mathbf{u}^* = (\pi^*(x_0, t_0), \dots, \pi^*(x(T-1), T-1))$$

where $x(t+1)^* = f[x(t)^*, \pi^*(x(t)^*, t), t]$ for all t .

The existence of an optimal policy states that knowledge of the current state is sufficient to determine the current input. Existence of such a policy vastly simplifies the optimization problem. However, not every generalized dynamic programming problem admits an optimal policy. The “Principle of Optimality” defines one class of optimization problems for which there exists an optimal policy.

Definition 3: We say an optimization problem, $G(t_0, x_0)$, of the Form (1) satisfies the principle of optimality if the following holds. For any s and t with $t_0 \leq t < s < T$, if $\mathbf{u}^* = (u(t), \dots, u(T-1))$ and $\mathbf{x}^* = (x(t), \dots, x(T))$ solve $G(t, x(t))$ then $\mathbf{v} = (u(s), \dots, u(T-1))$ and $\mathbf{h} = (x(s), \dots, x(T))$ solve $G(s, x(s))$.

The classical form of Dynamic programming algorithm, as originally defined in [20], can be used to solve indexed optimization problems of the Form (1). This algorithm has the advantage of computational complexity which is linear in T .

Dynamic Programming algorithms are most commonly used to solve the special class of indexed optimization problems $P(t_0, x_0)$ of the form

$$\min_{\mathbf{u}, \mathbf{x}} J_{t_0, x_0}(\mathbf{u}, \mathbf{x}) = \sum_{t=t_0}^{T-1} c_t(x(t), u(t)) + c_T(x(T)) \quad (2)$$

subject to: $x(t+1) = f[x(t), u(t), t]$, given $x(t_0) = x_0$

$x(t) \in X$ for $t = t_0 + 1, \dots, T$

$u(t) \in U$ for $t = t_0, \dots, T-1$

Note that $J_{T,x} = c_T(x)$. We will refer to $x(t_0) \in \mathbb{R}^n$ the initial state, $x(t) \in \mathbb{R}^n$ the state at time t and $u(t) \in \mathbb{R}^m$ the inputs at time t . J_{t_0, x_0} is the objective function, $c_t : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$ for $t = t_0, \dots, T-1$, $c_T : \mathbb{R}^n \rightarrow \mathbb{R}$ are given functions and $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{N} \rightarrow \mathbb{R}^n$ is a given vector field. The following lemma shows that this class of problems satisfies the principle of optimality.

Lemma 1: Any Problem of Form $P(t_0, x_0)$ in (2) satisfies the principle of optimality.

Proof: Suppose $\mathbf{u}^* = (u(t), \dots, u(T-1))$ and $\mathbf{x}^* = (x(t), \dots, x(T))$ solve $P(t, x(t))$ in (2). Now we suppose by contradiction that there exists some $s > t$ such that $\mathbf{v} = (u(s), \dots, u(T-1))$ and $\mathbf{h} = (x(s), \dots, x(T))$ do not solve $P(s, x(s))$. We will show that this implies that \mathbf{u}^* and \mathbf{x}^* do not solve $P(t, x)$ in (2), thus verifying the conditions of the Principle of Optimality. If \mathbf{v} and \mathbf{h} do not solve $P(s, x(s))$, then there exist feasible \mathbf{w}, \mathbf{z} such that $J_{s,x(s)}(\mathbf{v}, \mathbf{h}) < J_{s,x(s)}(\mathbf{w}, \mathbf{z})$. i.e.

$$\begin{aligned} J_{s,x(s)}(\mathbf{w}, \mathbf{z}) &= \sum_{t=s}^{T-1} c_t(z(t), w(t)) + c_T(z(T)) \\ &< \sum_{t=s}^{T-1} c_t(x(t), u(t)) + c_T(x(T)) \\ &= J_{s,x(s)}(\mathbf{v}, \mathbf{h}) \end{aligned}$$

Now consider the proposed feasible sequences $\hat{\mathbf{u}} = (u(t), \dots, u(s-1), w(s), \dots, w(T-1))$ and $\hat{\mathbf{x}} = (x(t), \dots, x(s-1), z(s), \dots, z(T-1))$. It follows:

$$\begin{aligned} J_{t,x(t)}(\hat{\mathbf{u}}, \hat{\mathbf{x}}) &= \sum_{k=t}^{s-1} c_k(x(k), u(k)) + \sum_{k=s}^{T-1} c_k(z(k), w(k)) + c_T(z(T)) \\ &< \sum_{k=t}^{s-1} c_k(x(k), u(k)) + \sum_{k=s}^{T-1} c_k(x(k), u(k)) + c_T(x(T)) \\ &= J_{t,x(t)}(\mathbf{u}^*, \mathbf{x}^*) \end{aligned}$$

which contradicts optimality of $\mathbf{u}^*, \mathbf{x}^*$. Therefore, this class of problems satisfies the principle of optimality. ■

Proposition 1: Consider the class of optimization problems $P(t_0, x_0)$ in (2). If we define $F(x, t) = J_{t,x}^*$, then the following hold for all $x \in X$.

$$F(x, t) = \inf_{u \in \Gamma_{t,x}} \{c_t(x, u) + F(f(x, u, t), t+1)\} \quad (3)$$

$$\forall t \in \{t_0, \dots, T-1\}$$

$$F(x, T) = c_T(x) \quad \forall x \in X$$

Proof:

Clearly $F(x, T) = c_T(x)$ for any x .

Now for any $x \in X$ and $t \in \{t_0, \dots, T-1\}$, suppose $\mathbf{u}^* = (u(t), \dots, u(T-1))$ and $\mathbf{x}^* = (x(t), \dots, x(T))$ solve $P(t, x)$. By the principle of optimality $\mathbf{v} = (u(t+1), \dots, u(T-1))$ and $\mathbf{h} = (x(t+1), \dots, x(T))$ solve $P(t+1, x(t+1))$. Therefore

$$F(x(t+1), t+1) = J_{x(t+1), t+1}^* = J_{x(t+1), t+1}(\mathbf{v}, \mathbf{h}). \quad (4)$$

We conclude that

$$\begin{aligned} F(x, t) &= J_{x,t}^* \\ &= J_{x,t}(\mathbf{u}^*, \mathbf{x}^*) \\ &= c_t(x, u(t)) + J_{x(t+1), t+1}(\mathbf{v}, \mathbf{h}) \\ &= c_t(x, u(t)) + F(f(x, u(t), t), t+1) \text{ using (4)} \\ &\geq \inf_{u \in \Gamma_{t,x}} \{c_t(x, u) + F(f(x, u, t), t+1)\} \end{aligned}$$

holds for all x and t .

Now we prove $F(x, t) \leq \inf_{u \in \Gamma_{t,x}} \{c_t(x, u) + F(f(x, u, t), t+1)\}$. For any $u \in \Gamma_{t,x}$, let $\mathbf{w}_u = \{w(t+1), \dots, w(T-1)\}$ and $\mathbf{h}_u = \{f(x, u, t), z(t+2), \dots, z(T)\}$ be feasible for $P(f(x, u, t), t+1)$. Then $\mathbf{v}_u = \{u, w(t+1), \dots, w(T-1)\}$ and $\mathbf{z}_u = \{x, f(x, u, t), z(t+2), \dots, z(T)\}$ are feasible for $P(t, x)$. Therefore,

$$\begin{aligned} F(x, t) &= J_{t,x}^* \leq J_{t,x}(\mathbf{v}_u, \mathbf{h}_u) \\ &= c_t(x, u) + J_{f(x, u, t), t+1}(\mathbf{w}_u, \mathbf{z}_u) \\ &\leq c_t(x, u) + F(f(x, u, t), t+1) \\ &\leq \inf_u \{c_t(x, u) + F(f(x, u, t), t+1)\} \end{aligned}$$

■

Note: Equation (3) is often referred to as Bellman's Equation and a function F which satisfies Bellman's equation is often referred to as a "cost to go" function. Prop. 1 shows that problems of the Form $P(t_0, x_0)$ admit a solution to Bellman's Equation which in turn indexes the optimal objective to the Problem. Furthermore, for problems $P(t_0, x_0)$, the solution to Bellman's equation can be obtained recursively backwards in time using a minimization on u . When x and u are discrete, the RHS of Eqn. 3 takes a number of finite values and minimization over these values is trivial. When the variables are continuous, finding a functional form for the minimization step is more challenging. In either case, a solution to Bellman's equation provides a state-feedback law or optimal policy as follows.

Corollary 1: Consider $P(t_0, x_0)$ in (2). Suppose $F(x, t)$ satisfies Equation (3) for $P(t_0, x_0)$, Then

$$\pi^*(x, t) = \arg \inf_{u \in \Gamma_{t,x}} \{c_t(x(t), u) + F(f(x(t), u, t), t+1)\}.$$

is an optimal policy.

Dynamic Programming with Supremum Terms In this paper we consider the special class of indexed optimization problem, $S(t_0, x_0)$. In contrast to problems of the form $P(t_0, x_0)$ in (1), class $S(t_0, x_0)$ has supremum (or maximum) terms in the objective. Specifically, these problems have the following form.

$$\min_{\mathbf{u}, \mathbf{x}} J_{t_0, x_0}(\mathbf{u}, \mathbf{x}) := \sum_{t=t_0}^{T-1} c_t(x(t), u(t)) + c_T(x(T)) + \sup_{t_0 \leq k \leq T} d_t(x(k)) \quad (5)$$

subject to: $x(t+1) = f[x(t), u(t), t]$

$x(0) = x_0$ given

$x(t) \in X$ for $t = t_0, \dots, T$

$u(t) \in U$ for $t = t_0, \dots, T-1$

Lemma 2: The class of optimization problems in (5) does not satisfy the principle of optimality.

Proof: We give a counterexample. For $h > 0$, we consider the following problem $S(0, 0)$:

$$\begin{aligned} \min_{\mathbf{u} \in \mathbb{R}^3, \mathbf{x} \in \mathbb{R}^4} \quad & \sum_{t=0}^2 c_t(u(t)) + \sup_{0 \leq k \leq 3} x(k) \\ \text{subject to: } & x(t+1) = x(t) + u(t), \quad x(0) = 0 \\ & 0 \leq x_t \leq h, \\ & u(t) \in \{-h, 0, h\} \end{aligned}$$

TABLE I

THIS TABLE SHOWS THE CORRESPONDING COST OF EACH FEASIBLE POLICY USED IN THE COUNTER EXAMPLE IN LEMMA 1

feasible \mathbf{u}	objective value	feasible \mathbf{u}	objective value
$(0, 0, 0)$	0	$(h, 0, -h)$	$h/2$
$(0, 0, h)$	$h/2$	$(h, 0, 0)$	0
$(0, h, 0)$	$2h$	$(h, -h, 0)$	$-h$
$(0, h, -h)$	$(5/2)h$	$(h, -h, h)$	$-(3/2)h$

Where here we define $c_0(u(0)) = -u(0)$, $c_1(u(1)) = u(1)$, $c_2(u(2)) = -u(2)/2$.

Since $\mathbf{u} \in \{-h, 0, h\}^3$, there are 27 input sequences, only 8 of which are feasible. In Table I, we calculate the objective value of each feasible input sequence and deduce the optimal input is $\mathbf{u} = (h, -h, h)$. Now suppose we follow this input sequence until $t = 2$ yielding $x(2) = 0$. Now we examine the problem $S(2, x(2))$.

$$\begin{aligned} & \min_{\mathbf{u} \in \mathbb{R}^n \times \mathbb{R}^p} c_2(u(2)) + \sup_{2 \leq k \leq 3} x(k) \\ & \text{subject to: } x(t+1) = x(t) + u(t), \quad x(2) = 0 \\ & 0 \leq x(t) \leq h, \\ & u(t) \in \{-h, 0, h\} \end{aligned}$$

For this sub-problem, there are two feasible inputs: $u(3) \in \{-h, 0\}$. Of these, the latter is optimal (objective value $h/2$ vs 0). Thus we see that although $\mathbf{u} = \{h, -h, h\}$ and $\mathbf{x} = \{0, h, 0, h\}$ solve $S(0, 0)$, $\mathbf{v} = \{h\}$ and $\mathbf{h} = \{0, h\}$ do not solve $S(2, 0)$. ■

III. SOLUTION METHODOLOGY: AUGMENTED DYNAMIC PROGRAMMING

In this section we will define what a forward separable objective function is and later show that the supremum is an example of such a function. We will show that for dynamic programming problems with a forward separable objective function, augmenting the state variables allows us to use standard dynamic programming techniques to solve the problem.

Definition 4 ([21]): The function $J(\mathbf{u}, \mathbf{x})$ is said to be forward separable if there exists functions $\phi_0(x, u)$, $\phi_T(x, \phi_{T-1})$, and $\phi_i(x, u, \phi_{i-1})$ for $i = 1, \dots, T-1$ such that

$$J(\mathbf{u}, \mathbf{x}) = \phi_T(x(T), \phi_{T-1}[x(T-1), u(T-1), \phi_{T-2}\{\dots, \phi_2\{x(2), u(2), \phi_1\{x(1), u(1), \phi_0\{x(0), u(0)\}\}\}, \dots, \}]) \quad (6)$$

where $\phi_t : \mathbb{R}^n \times \mathbb{R}^p \times \mathbb{R}^q \rightarrow \mathbb{R}^q$, for $t = 1, \dots, T-1$ and $\phi_T : \mathbb{R}^n \times \mathbb{R}^q \rightarrow \mathbb{R}$, $\phi_0 : \mathbb{R}^n \times \mathbb{R}^p \rightarrow \mathbb{R}^q$.

Clearly, any objective function of the form

$$J(\mathbf{u}, \mathbf{x}) = \sum_{t=0}^{T-1} c_t(u(t), x(t)) + c_T(x(T))$$

is forward separable using $\phi_0(x, u) = c_0(x, u)$, $\phi_T(x, \phi_{T-1}) = c_T(x) + \phi_{T-1}$ and

$$\phi_i(x, u, \phi_{i-1}) = c_i(x, u) + \phi_{i-1} \quad \text{for } i = 1, \dots, T-1$$

In addition, it can be shown that the sum of any number of forward separable functions is forward separable. For example, let $J_1(\mathbf{u}, \mathbf{x})$ and $J_2(\mathbf{u}, \mathbf{x})$ be forward separable with associated $\phi_i = g_i$ and $\phi_i = h_i$, respectively. Then $J_1 + J_2$ is forward separable with

$$\phi_i(x, u, \phi_{i-1}) = \begin{bmatrix} \phi_i^1(x, u, \phi_{i-1}) \\ \phi_i^2(x, u, \phi_{i-1}) \end{bmatrix} = \begin{bmatrix} g_i(x, u, \phi_{i-1}^1) \\ h_i(x, u, \phi_{i-1}^2) \end{bmatrix} \quad (7)$$

and

$$\phi_T(x, u, \phi_{T-1}) = g_T(x, u, \phi_{T-1}^1) + h_T(x, u, \phi_{T-1}^2).$$

Clearly,

$$\phi_0(x, u) = \begin{bmatrix} \phi_0^1(x, u) \\ \phi_0^2(x, u) \end{bmatrix} = \begin{bmatrix} g_0(x, u) \\ h_0(x, u) \end{bmatrix}.$$

We now show that the supremum (maximum) function is forward separable.

Lemma 3:

$$J(\mathbf{u}, \mathbf{x}) = \max\left\{\sup_{0 \leq k \leq T-1} \{c_k(u(k), x(k))\}, c_T(x(T))\right\}$$

is a forward separable objective function.

Proof:

$$\begin{aligned} J(\mathbf{u}, \mathbf{x}) &= \max\left\{\sup_{0 \leq k \leq T-1} \{c_k(u(k), x(k))\}, c_T(x(T))\right\} \\ &= \max\{c_T(x(T)), \max\{c_{T-1}(u(T-1), x(T-1)), \dots \\ &\quad \max\{\dots, \max\{c_1(u(1), x(1)), \max\{c_0(u(0), x(0))\}\}, \dots\}\} \end{aligned}$$

so that

$$\begin{aligned} \phi_i(x, u, \phi_{i-1}) &= \max(c_i(x, u), \phi_{i-1}), \quad \phi_0(x, u) = c_0(x, u), \\ \phi_T(x, \phi_{T-1}) &= \max(c_T(x), \phi_{T-1}) \end{aligned}$$

■

A. Forward Separable Dynamic Programming

We may now define the class of indexed forward separable problems $H(t_0, x_0)$ so that H is of class G , but not of class P and has the form:

$$\begin{aligned} & \min_{\mathbf{u}, \mathbf{x}} J_{t_0, x_0}(\mathbf{u}, \mathbf{x}) \\ & \text{subject to: } x(t+1) = f[x(t), u(t), t] \\ & x(0) = x_0 \\ & x(t) \in X \subset \mathbb{R}^n \text{ for } t = 1, \dots, T \\ & u(t) \in U \subset \mathbb{R}^p \text{ for } t = 0, \dots, T-1 \end{aligned} \quad (8)$$

where J_{t_0, x_0} is forward separable with associated ϕ_i . For every instance of a forward separable dynamic programming problem $H(t_0, x_0)$, we may associate a new optimization problem $A(t_0, x_0)$, which is equivalent to $H(t_0, x_0)$ in a certain sense and which satisfies the principle of optimality. $A(t_0, x_0)$ is defined as follows.

$$\begin{aligned}
& \min_{\mathbf{u}} L_{t_0, x_0}(\mathbf{u}, \mathbf{x}) = z_2(T+1) \\
& \text{subject to: } \begin{bmatrix} z_1(t+1) \\ z_2(t+1) \end{bmatrix} = \begin{bmatrix} f(z_1(t), u(t)) \\ \phi_t(z_1(t), u(t), z_2(t)) \end{bmatrix} \quad 1 < t < T \quad (9) \\
& \begin{bmatrix} z_1(1) \\ z_2(1) \end{bmatrix} = \begin{bmatrix} f(z_1(0), u(0)) \\ \phi_0(z_1(0), u(0)) \end{bmatrix}, \quad \begin{bmatrix} z_1(T+1) \\ z_2(T+1) \end{bmatrix} = \begin{bmatrix} z_1(T) \\ \phi_T(z_1(T), z_2(T)) \end{bmatrix} \\
& \begin{bmatrix} z_1(0) \\ z_2(0) \end{bmatrix} = \begin{bmatrix} x_0 \\ 0 \end{bmatrix} \\
& z_1(t) \in X \text{ for } t = 1, \dots, T \\
& u(t) \in U \text{ for } t = 1, \dots, T
\end{aligned}$$

Where the solution to $H(t_0, x_0)$ can be recovered as $x(t) = z_1(t)$.

Lemma 4: Suppose $J_{t_0, x_0}(\mathbf{u}, \mathbf{x})$ is forward separable with associated ϕ_i . Then $J_{t_0, x_0}^* = L_{t_0, x_0}^*$. Furthermore, suppose \mathbf{u} and \mathbf{x} solve $H(t_0, x_0)$ and \mathbf{w} and \mathbf{z} solve $A(t_0, x_0)$. Then $\mathbf{u} = \mathbf{w}$ and $x(t) = z_1(t)$ for all t .

Proof: Suppose \mathbf{w} and \mathbf{z} solve $A(t_0, x_0)$. First we show that \mathbf{w} and \mathbf{z}_1 are feasible for $H(t_0, x_0)$. Clearly $w(t) \in U$ for all t and if we let $\mathbf{u} = \mathbf{w}$ then $x(0) = x_0$ and $x(t+1) = f[x(t), u(t), t]$ for all t . Since likewise $z_1(0) = x_0$ and $z_1(t+1) = f[z_1(t), u(t), t]$, we have $x(t) = z_1(t) \in X$ for all t . Hence \mathbf{u} and $\mathbf{x} = \mathbf{z}_1$ are feasible for $H(t_0, x_0)$. Likewise, if \mathbf{u} and \mathbf{x} solve $H(t_0, x_0)$, then if we let $\mathbf{w} = \mathbf{u}$ and $\mathbf{z}_1 = \mathbf{x}$ and define $z_2(t+1) = \phi_t(z_1(t), u(t), z_2(t))$, $z_2(1) = \phi_0(z_1(0), u(0))$, $z_2(0) = 0$, then \mathbf{w} and \mathbf{z} are feasible. Furthermore, in both cases, if we examine the objective value

$$\begin{aligned}
J(\mathbf{u}, \mathbf{x}) &= \phi_T(z_1(T), \phi_{T-1}[z_1(T-1), w(T-1), \phi_{T-2}\{\dots, \\
&\quad \phi_2\{z_1(2), w(2), \phi_1\{z_1(1), w(1), \phi_0\{z_1(0), w(0)\}\}\}, \dots, \}]).
\end{aligned}$$

However, we now observe

$$\begin{aligned}
z_2(T+1) &= \phi_T(z_1(T), z_2(T)) \\
z_2(T) &= \phi_{T-1}(z_1(T-1), u(T-1), z_2(T-1)) \\
&\vdots \\
z_2(2) &= \phi_1(z_1(1), u(1), z_2(1)) \\
z_2(1) &= \phi_0(z_1(0), u(0)).
\end{aligned}$$

Hence we have

$$\begin{aligned}
L(\mathbf{w}, \mathbf{z}) &= z_1(T+1) \\
&= \phi_T(z_1(T), u(T), \phi_{T-1}(z_1(T-1), u(T-1), \phi_{T-2}(\dots, \\
&\quad \phi_1(z_1(1), u(1), z_2(1), \phi_0(z_1(0), u(0)))))) \\
&= J(\mathbf{u}, \mathbf{x}).
\end{aligned}$$

Hence if \mathbf{w} and \mathbf{z} solve $A(t_0, x_0)$ with objective $L_{t_0, x_0}^* = z_2(T+1)$, then \mathbf{w} and \mathbf{z}_1 solve $H(t_0, x_0)$ with objective value $J_{t_0, x_0}^* = L_{t_0, x_0}^* = z_2(T+1)$. ■

Proposition 2: The augmented optimization problem $A(t_0, x_0)$ in (9) satisfies the Principle of Optimality and the Bellman equation (3).

Proof: $A(t_0, x_0)$ is a special case of $P(t_0, x_0)$ where $c_i = 0$ for $i \neq T$. ■

To understand the augmented approach intuitively, we note that dynamic programming breaks a multi-period planning problem into simpler optimization problems at each stage. However, for non-separable problems, to make the correct decision at each stage we need historical data. In this context, the extra augmented state contains that part of the history necessary to make the correct decision at the present time.

Corollary 2: $S(t_0, x_0)$ is a special case of $H(t_0, x_0)$.

Proof: Consider the objective function from Problem $S(t_0, x_0)$ as

$$J_{t_0, x_0}(\mathbf{u}, \mathbf{x}) = \sum_{t=t_0}^{T-1} c_t(x(t), u(t)) + c_T(x(T)) + \sup_{t_0 \leq k \leq T} d_t(x(k)).$$

Now this is the sum of two forward separable functions. As per the previous discussion (7), then, we define

$$g_i(x, u, \phi_{i-1}^1) = c_i(x, u) + \phi_{i-1}^1 \quad \text{for } i = 1, \dots, T-1$$

$$g_0(x, u) = c_0(x, u), \quad g_T(x, \phi_{T-1}^1) = c_T(x) + \phi_{T-1}^1$$

and

$$\begin{aligned}
h_i(x, u, \phi_{i-1}^2) &= \max(d_i(x, u), \phi_{i-1}^2), \quad h_0(x, u) = d_0(x, u), \\
h_T(x, \phi_{T-1}^2) &= \max(d_T(x), \phi_{T-1}^2)
\end{aligned}$$

Then

$$\phi_i(x, u, \phi_{i-1}) = \begin{bmatrix} \phi_i^1(x, u, \phi_{i-1}) \\ \phi_i^2(x, u, \phi_{i-1}) \end{bmatrix} = \begin{bmatrix} g_i(x, u, \phi_{i-1}^1) \\ h_i(x, u, \phi_{i-1}^2) \end{bmatrix}$$

$$\phi_T(x, u, \phi_{T-1}) = g_T(x, u, \phi_{T-1}^1) + h_T(x, u, \phi_{T-1}^2),$$

and

$$\phi_0(x, u) = \begin{bmatrix} \phi_0^1(x, u) \\ \phi_0^2(x, u) \end{bmatrix} = \begin{bmatrix} g_0(x, u) \\ h_0(x, u) \end{bmatrix}$$

establish forward separability of J_{t_0, x_0} as per (6). ■

The ϕ_i specified in the proof of this Corollary define an instance of problem $H(t_0, x_0)$, which was shown to be equivalent to a class of optimization problems $A(t_0, x_0)$ by Lemma 4. Since problems of class $A(t_0, x_0)$ satisfy the principle of optimality, they can be solved using dynamic programming and their solution yields a solution to the original Problem $S(t_0, x_0)$. In the following section, we will apply this technique to optimal battery scheduling in the presence of demand charges.

IV. APPLICATION TO THE ENERGY STORAGE PROBLEM

In this section, we apply the augmented dynamic programming methodology to optimal scheduling of batteries in the presence of demand charges. We first propose a simple model for the dynamics of the battery storage. We then formulate the objective function using electricity pricing plans which include demand charges. We see that the system described becomes an optimization problem of the form $H(0, e_0)$ (8).

A. Battery Dynamics

We will model the energy stored in the battery by the difference equation:

$$e(k+1) = \alpha(e(k) + \eta u(k)\Delta t) \quad (10)$$

Where $e(k)$ denotes the energy stored in the battery at time step k , α is the bleed rate of the battery, η is the efficiency of the battery, $u(k)$ denotes the charging/discharging (+/-) at time step k and Δt is the amount of time passed between each time step. Moreover we denote the maximum charge and discharge rate by \bar{u} and \underline{u} respectively. Thus we have the constraint that $u(k) \in [\underline{u}, \bar{u}] := U$ for all k . Similarly we also add the constraint $e(k) \in [\underline{e}, \bar{e}] := X$ for all k where \underline{e} and \bar{e} are the capacity constraints of the battery (typically $\underline{e} = 0$).

B. The objective function

Let us denote $q(k)$ to be the power supplied by the grid at time step k .

$$q(k) = q_a(k) - q_s(k) + u(k) \quad (11)$$

where $q_a(k)$ is the power consumed by HVAC/appliances at time step k and $q_s(k)$ is the power supplied by solar photovoltaics at time step k . For now, it is assumed that both are known apriori.

To define the cost of electricity we divide the day $t \in [0, T]$ into on-peak and off-peak periods. We define an off peak period starting from 12am till t_{on} and t_{off} till 12am. We define an on-peak period between t_{on} till t_{off} . The Time-of-Use (TOU, \$ per kWh) electricity cost during on-peak and off-peak is denoted by p_{on} and p_{off} respectively. We further simplify this as $p_k = p_{\text{on}}$ if $k \in T_{\text{on}}$ and $p_k = p_{\text{off}}$ if $k \in T_{\text{off}}$ where T_{on} and T_{off} are the on-peak and off-peak hours, respectively. These TOU charges define the first part of the objective function as:

$$\begin{aligned} J_E(\mathbf{u}, \mathbf{e}) &= p_{\text{off}} \sum_{k=0}^{t_{\text{on}}-1} q(k)\Delta t + p_{\text{on}} \sum_{k=t_{\text{on}}}^{t_{\text{off}}-1} q(k)\Delta t + p_{\text{off}} \sum_{k=t_{\text{off}}}^T q(k)\Delta t \\ &= \sum_{k \in [0, T]} p_k (q_a(k) - q_s(k) + u(k))\Delta t \\ &= \sum_{k \in [0, T]} p_k (q_a(k) - q_s(k))\Delta t + \sum_{k \in [0, T]} p_k u(k)\Delta t \end{aligned}$$

Where the daily terminal timestep is $T = 24/\Delta t$. Clearly, only the second term in this objective function is significant for the purposes of optimization.

We also include a demand charge, which is a cost proportional to the maximum rate of power taken from the grid during on-peak times. This cost is determined by p_d which is the price in \$ per kW. Thus it follows the demand charge will be:

TABLE II
LIST OF CONSTANT VALUES (PRICES CORRESPOND TO SALT RIVER PROJECT E21 PRICE PLAN)

Constant	Value	Constant	Value
α	0.999791667 (W/h)	t_{off}	41
η	0.92 (%)	p_{on}	0.0633×10^{-3} (\$/KWh)
\bar{u}	4000 (Wh)	p_{off}	0.0423×10^{-3} (\$/KWh)
\underline{u}	-4000 (Wh)	p_d	3.364 (\$/KWh)
\bar{e}	8000 (Wh)	Δt	0.5 (h)
t_{on}	27		

$$\begin{aligned} J_D(\mathbf{u}, \mathbf{e}) &= p_d \sup_{k \in \{t_{\text{on}}, \dots, t_{\text{off}}-1\}} q(k) \\ &= p_d \sup_{k \in \{t_{\text{on}}, \dots, t_{\text{off}}-1\}} \{q_a(k) - q_s(k) + u(k)\} \end{aligned}$$

C. 24 hr Optimal Residential Battery Storage Problem

We may now define the problem of optimal battery scheduling in the presence of demand and Time-of-Use charges, denoted $D(0, e_0)$.

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{e}} \{ & J_E(\mathbf{u}, \mathbf{e}) + J_D(\mathbf{u}, \mathbf{e}) \} \text{ subject to} \\ & e(k+1) = \alpha(e(k) + \eta u(k)\Delta t) \text{ for } k = 0, \dots, T \\ & e(k) \in X \text{ for } k = 0, \dots, T \\ & u(k) \in U \text{ for } k = 0, \dots, T \\ & e_0 = e_0 \end{aligned}$$

Where recall $U := [\underline{u}, \bar{u}]$ and $X := [\underline{e}, \bar{e}]$.

Proposition 3: Problem $D(0, e_0)$ is a special case of $S(t_0, x_0)$

Proof: Let $c_i = p_i(q_a(i) - q_s(i) + u(i))\Delta t$

$$d_i = \begin{cases} p_d(q_a(k) - q_s(k) + u_k) & k \in T_{\text{on}} \\ 0 & \text{otherwise.} \end{cases}$$

We conclude that our algorithmic approach to forward separable dynamic programming can be applied to this problem as per Corollary 2. That is, it can be represented as an augmented dynamic programming problem of Form $A(t_0, x_0)$.

V. NUMERICAL IMPLEMENTATION

To illustrate our approach to generalized dynamic programming, we use solar and usage data obtained by local utility Salt River Project in Tempe, AZ. We also use pricing data from SRP and battery data obtained for the Tesla Powerwall. As is standard practice, for implementation, we used a discrete input and state space. The results of the simulation are shown in Fig. V. These results show a slight improvement in accuracy over results obtained based on the approach to a similar problem in [18] (approximately \$0.98 savings).

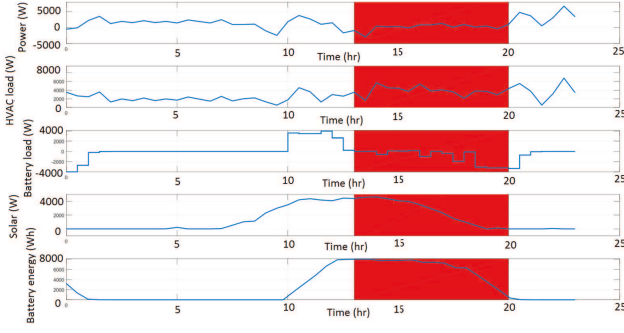


Fig. 1. The trajectory the algorithm produces for randomly generated stochastic solar data. The supremum of the power is 1.66(kw) and the cost is \$64.9889.

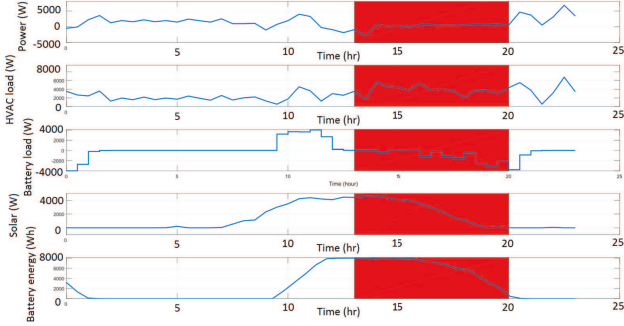


Fig. 2. The trajectory the algorithm produces for deterministic solar data. The supremum of the power is 0.7033(kw) and the cost is \$46.389.

VI. USING A STOCHASTIC MODEL

To show that this approach can also be extended to stochastic dynamic programming and to evaluate the effect of stochastic uncertainty on battery scheduling, we identified a Gauss-Markov model of solar generation based on SRP data. We then used a trivial extension of problem $A(t_0, s_0)$ to problems with stochastic disturbances.

A. Solar Generation Model

Our approach to modeling the dynamics of load following for a given subset of data is to model solar irradiance directly as a primary variable along with other possible correlated variables such as temperature or 2-hr Deltas in pressure. Specifically, we take time-series data of these quantities, denoted $\mathbf{W}(t)$ and normalize this data as

$$w_i(t) = \frac{W_i(t) - \mu_i(t)}{\sigma_i(t)}$$

Where $\mu_i(t)$ is the average historic and clear-sky mean of the variable W_i at time step t and $\sigma_i(t)$ is the standard deviation of variable W_i at time step t .

The generating process is then given by:

$$\mathbf{w}(t) = \mathbf{A}\mathbf{w}(t-1) + \mathbf{B}\boldsymbol{\varepsilon}(t-1) \text{ for } t = 1, \dots, T$$

$$\text{where } \mathbf{w}(t) \in \mathbb{R}^3, \mathbf{w}(0) = \mathbf{0}$$

$$\boldsymbol{\varepsilon}(t) \sim \mathcal{N}(\mathbf{0}, \Sigma), \Sigma_{i,j} = \delta_{i,j} := \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

Where the matrices \mathbf{A} and \mathbf{B} are chosen to preserve the lag 0 and lag 1 cross-correlations seen in the collected data. Specifically, we can compute these matrices as ([22])

$$\mathbf{A} = \mathbf{M}_1 \mathbf{M}_0^{-1} \quad \mathbf{B} \mathbf{B}^T = \mathbf{M}_0 - \mathbf{M}_1 \mathbf{M}_0^{-1} \mathbf{M}_1^T$$

Where \mathbf{M}_i is the i -lag cross correlation matrix. So $(\mathbf{M}_i)_{m,n} = \rho_i(m,n)$ where $\rho_i(m,n)$ is the cross-correlation coefficient between variables m and n with variable n lagged by i time steps. Then, adding back in the mean and deviation, we obtain the power supplied by solar at time step k as

$$q_s(k) = w_1(k)\sigma_1(k) + \mu_1(k)$$

B. Augmented Stochastic Dynamic Programming

We now define a class of Stochastic Dynamic Programming problems, $T(t_0, x_0)$ of the form

$$\min_{\mathbf{u}, \mathbf{x}} J_{t_0, x_0}(\mathbf{u}, \mathbf{x}) = \mathbb{E} \left(\sum_{t=t_0}^{T-1} c_t(x(t), u(t)) + c_T(x(T)) \right) \quad (12)$$

subject to: $x(t+1) = f[x(t), u(t), t; v(t)]$, given $x(t_0) = x_0$

$x(t) \in X$ for $t = t_0 + 1, \dots, T$

$u(t) \in U$ for $t = t_0, \dots, T-1$

$v(t) \sim \mathcal{N}(\mathbf{0}, \Sigma), \quad \Sigma_{i,j} = \delta_{i,j} \quad (13)$

As shown in [20], A stochastic version of the Bellman Equation can be used to solve Stochastic Dynamic Programming problems of the Form $T(t_0, x_0)$. Specifically, suppose that F satisfies $F(x, T) = c_T(x)$ and

$$F(x, t) = \inf_u \{ c_t(x, u) + \mathbb{E}_v [F(f[x, u, t; v], t+1) | x, u] \}. \quad (14)$$

Then for problem $T(t_0, x_0)$, $F(x, t) = J_{t,x}^*$ and $\pi^*(x) = \inf_u \{ c_t(x, u) + \mathbb{E}_v [F(f[x, u, t; v], t+1)] \}$ defines an optimal policy.

Stochastic Battery Scheduling We now modify Problem $D(0, e_0)$ to give a stochastic version of the battery scheduling problem

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{e}} \mathbb{E}(\{J_E(\mathbf{u}, \mathbf{e}) + J_D(\mathbf{u}, \mathbf{e})\}) \text{ subject to} \\ e(k+1) = \alpha(e(k) + \eta u(k)\Delta t) \text{ for } k = 0, \dots, T \\ w(k+1) = \mathbf{A}w(k) + \mathbf{B}\boldsymbol{\varepsilon}(k) \text{ for } k = 0, \dots, T \\ e(k) \in X \text{ for } k = 0, \dots, T \\ u(k) \in U \text{ for } k = 0, \dots, T \\ e_0 = e0, \quad \boldsymbol{\varepsilon}(k) \sim \mathcal{N}(\mathbf{0}, \Sigma) \end{aligned}$$

To solve the stochastic version of $D(0, e_0)$, we augment to obtain a stochastic version of Problem $A(t_0, x_0)$, which is a special case of $T(t_0, x_0)$, which then admits a solution using the stochastic version of Bellman's equation.

C. Implementation of the Stochastic Algorithm

The primary challenge with implementation is computing the expectation in Bellman's equation (14). Specifically, if ϕ is the pdf of $\mathbf{v}(t)$, we must compute

$$\mathbb{E}_v [F(f[x, u, t; v], t+1) | x, u] = \int F(f(x, u, t; v), t+1) \phi(v) dv$$

To numerically integrate this function, we discretize u , x and v so that the integral becomes a sum where $\phi(v_i)$ is a weighted sample from the normal distribution. The results of this algorithm are illustrate in Figure V using the parameter values from Table II. The solar data generated from this run were then used as input to the deterministic algorithm in order to compare performance. As expected, the deterministic case performs better than the stochastic case.

VII. CONCLUSION

In this paper we have proposed a generalized formulation of the dynamic programming problem and shown that if the objective function is forward separable, these problems may be solved using an equivalent augmented dynamic programming approach. Furthermore, we have shown that the problem of optimal scheduling of battery storage in the presence of combined demand and time-of-use charges is a special case of this class of forward separable dynamic programming problems. We have further extended these results to stochastic dynamic programming with a forward separable objective. The proposed algorithms were demonstrated on a battery scheduling problem using first a deterministic and then Gauss-Markov model for solar generation and load.

REFERENCES

- [1] Solar Energy Power Association, "SEPA comments on utility investments in distributed solar companies," *press release*, 2013.
- [2] L. Sherwood, "U.S. solar market trends 2012," Interstate Renewable Energy Council, Tech. Rep., July 2013.
- [3] J. J. Conti, "Annual energy outlook 2014 with projections to 2040," *US Energy Information Administration (EIA), Independent statistics & Analysis*, 2014.
- [4] T. Shear, "Today in energy: February archive," *US Energy Information Administration (EIA), Independent statistics & Analysis*, 2014.
- [5] S. R. Project, "Standard electric price plans," November 2015.
- [6] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, no. 1, pp. 18–28, 2010.
- [7] A. Mohd, E. Ortjohann, A. Schmelter, N. Hamsic, and D. Morton, "Challenges in integrating distributed energy storage systems into future smart grid," in *IEEE International Symposium on Industrial Electronics*, 2008, pp. 1627–1632.
- [8] B. Dunn, H. Kamath, and J.-M. Tarascon, "Electrical energy storage for the grid: A battery of choices," *Science*, vol. 334, no. 6058, pp. 928–935, 2011.
- [9] —, "Electrical energy storage for the grid: a battery of choices," *Science*, vol. 334, no. 6058, pp. 928–935, 2011.
- [10] M. C. Bozchalui, S. A. Hashmi, H. Hassen, C. A. Cañizares, and K. Bhattacharya, "Optimal operation of residential energy hubs in smart grids," *IEEE Transactions on Smart Grid*, vol. 3, no. 4, pp. 1755–1766, 2012.
- [11] T. Nagai, "Optimization method for minimizing annual energy, peak energy demand, and annual energy cost through use of building thermal storage/discussion," *ASHRAE Transactions*, vol. 108, p. 43, 2002.
- [12] G. P. Henze, M. J. Brandemuehl, C. Felsmann, H. Cheng, A. R. Florita, and C. E. Waters, "Optimization of building thermal mass control in the presence of energy and demand charges," *ASHRAE Transactions*, vol. 114, no. 2, 2008.
- [13] C. Lu, H. Xu, X. Pan, and J. Song, "Optimal sizing and control of battery energy storage system for peak load shaving," *Energies*, vol. 7, no. 12, pp. 8396–8410, 2014.
- [14] D. Maly and K. Kwan, "Optimal battery energy storage system (bess) charge scheduling with dynamic programming," *IEE Proceedings-Science, Measurement and Technology*, vol. 142, no. 6, pp. 453–458, 1995.
- [15] J. Cai, J. E. Braun, D. Kim, and J. Hu, "General approaches for determining the savings potential of optimal control for cooling in commercial buildings having both energy and demand charges," *Science and Technology for the Built Environment*, vol. 22, no. 6, pp. 733–750, 2016.
- [16] A. Zeinalzadeh and V. Gupta, "Minimizing risk of load shedding and renewable energy curtailment in a microgrid with energy storage," *arXiv preprint arXiv:1611.08000*, 2016.
- [17] R. Kamyar and M. M. Peet, "Multi-objective dynamic programming for constrained optimization of non-separable objective functions with application in energy storage," in *Decision and Control (CDC), 2016 IEEE 55th Conference on*. IEEE, 2016, pp. 5348–5353.
- [18] —, "Optimal thermostat programming and optimal electricity rates for customers with demand charges," in *American Control Conference (ACC), 2015*. IEEE, 2015, pp. 4529–4535.
- [19] D. Li and Y. Y. Haimes, "Extension of dynamic programming to non-separable dynamic optimization problems," *Computers & Mathematics with Applications*, vol. 21, no. 11-12, pp. 51–56, 1991.
- [20] R. E. Bellman, R. E. Kalaba, and T. Teichmann, "Dynamic programming," *Physics Today*, vol. 19, p. 98, 1966.
- [21] D. Li and Y. Y. Haimes, "The envelope approach for multiobjective optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 6, pp. 1026–1038, 1987.
- [22] C. W. Richardson, "Stochastic simulation of daily precipitation, temperature, and solar radiation," *Water resources research*, vol. 17, no. 1, pp. 182–190, 1981.