Extensions of the Dynamic Programming Framework: Battery Scheduling, Demand Charges, and Renewable Integration

Morgan Jones, Matthew M. Peet

***Abstract*—In this paper, we consider dynamic programming problems with non-separable objective functions. We 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 by extending the definition of the principle of optimality to problems driven by random variables. We then apply the resulting algorithms 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

Many problems in engineering and economics involve discrete time processes coupled with decision variables and an objective function. These optimization problems are commonly solved using Dynamic Programming (DP) [1]. DP is a class of algorithms that break down complex optimization problems into simpler sequential subproblems, each of which is solved using Bellman's Equation. For DP to work, however, we require that the optimization problem satisfies the *principle of optimality*; from any point on an optimal trajectory, the remaining portion of the optimal trajectory is also optimal for the problem initiated at that point [2]. DP problems commonly have an additively separable objective function of the form $J(\mathbf{u},\mathbf{x}) = \sum_{t=0}^{T-1} c_t(x(t),u(t)) + c_T(x(T))$. Problems of this form can be shown to satisfy the principle of optimality. However in many problems of practical interest we find non-additively separable objective functions. For example, if the objective is of the form $J(\mathbf{u},\mathbf{x}) = \max_{t_0 \le k \le T} d_k(x(k))$ then the problem does not satisfy the principle of optimality. In this paper we propose a general method for solving optimization problems with non-separable objective functions by constructing equivalent optimization problems with additively separable objective functions. Such reformulated problems then satisfy the principle of optimality and can therefore be solved using Bellman's Equation.

To generalize our methodology to stochastic DP we propose an extension of the definition of the *principle of optimality* to problems that involve random variables. As discussed in [3] such an extension is non trivial. Inspired by [4], we construct probability measures on the sets the state variable can take at each time stage induced by the underlying random variables, we then propose a *stochastic principle of optimality*; we say a stochastic problem satisfies the *principle of optimality* if from any point on a trajectory followed using the optimal policy, π, the policy π is also optimal for the problem initiated from that point with probability one. In economic theory a stochastic optimization problem that satisfies the principle of optimality is often called time consistent. In the deterministic setting this was first explored in [5]. The stochastic setting was further explored in [6] and [7]. This paper presents a rigorous proof that all DP problems with additively separable objective function satisfy the principle of optimality.

Dynamic programming for problems which do not satisfy the principle of optimality has received relatively 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 [8] which considered the use of multi-objective optimization in the case where the objective function is "backward separable". Our approach differs from [8] as we only consider a class of "forward separable" objective functions. In this paper we show that almost any objective function is forward separable in a certain sense and that for such problems there exists an additively separable augmented-state dynamic programming problem that satisfies the principle of optimality and from which solutions to the original forward separable problem can be recovered - See Section III. However, the resulting augmented-state dynamic programming problem has a higher dimensional state space than the original optimization problem - an issue that can potentially render the augmented problem intractable due to the "curse of dimensionality". For this reason, we propose a complexity metric for the forward separable representation and show that in certain cases the dimensionality of the augmented system does not significantly exceed the dimensionality of the original problem - a case we refer to as *Naturally Forward Separable* (NFS).

Note that the used of state augmentation has previously been used in the context of DP. For example, in: [10] where a Kalman filtering approach for force identification was presented and augmentation was used to account for the inherent error in the state variables; [11] where the state space was extended to account for noise correlations with previous disturbances. In [12] and [13] state augmentation was applied to a DP problem with non-separable objective function of the form $J(\mathbf{u},\mathbf{x}) = U(\sum_{t=0}^{T-1} c_t(x(t),u(t)))$. Finally, in [9], augmented states were mentioned as an approach to solving problems with non-separable variance type objective functions; however this method was ultimately rejected due to computational intractability.

The methods proposed in this

In this paper we therefore consider using an augmented state method, make it rigorous, and extend it to a general class

M. Jones is with the School for the Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, 85298 USA. e-mail: morgan.c.jones@asu.edu

M. Peet is with the School for the Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, 85298 USA. e-mail: mpeet@asu.edu

of NFS objective functions including both variance type and maximum type functions.

In practice it is rare to be able to analytically solve Bellman's Equation. Therefore, once the augmented-state DP problem is formulated we propose a map to an approximated DP problem that can be analytically solved. Using the optimal solution from the approximated DP problem a feasible policy for the original problem is then constructed. Our approximation scheme is based on the discretization of the state and input space. However, such schemes are known to suffer from the "curse of dimensionality"; which is unfortunate as state augmentation increases the state space dimension. Alternative schemes for numerical implementation, such as [14], claim to overcome the "curse of dimensionality" by using probabilistic techniques. Unfortunately the convergence of such algorithms can only be given in probability or expectation. We have therefore chosen to base our discretization method on the works of [15] and [16] that both give deterministic convergence rates.

In this paper we will apply our augmented-state dynamic programming methods to battery scheduling optimization problems. The use of battery storage has been well documented in the literature [17] and in particular, there have been several results on the optimal use of batteries for residential customers [18], [19], [20], [21]. Within this literature, there are relatively few results which include demand charges. Of those which do treat demand charges, we mention [22] which proposes a heuristic form of dynamic programming, and the recent work in [23], wherein the optimization problem is broken down into several agents, and a Lagrangian approach is used to perform the optimization. Furthermore, in [24] 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 [25] and, in addition, the optimal use of building mass for energy storage was considered in [26], 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 maximum 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 maximum terms in the objective do not satisfy the *principle of optimality* and thus recursive solution of the Bellman equation ([1]) does not yield an optimal policy.

Part of this work was presented at IEEE 56th Annual Conference on Decision and Control (CDC) 2017 [27]. This journal version contains new and significant contributions when compared to the conference version. Namely, a complexity metric for the representation of forward separable functions is proposed; we define a case of objective functions, called naturally forward separable functions, where we argue the use of state augmentation is tractable; we illustrate how to numerically solve DP problems once state augmentation is done; we compare the solution of a problem found in [8] with our proposed state augmentation method; we propose a definition of the principle of optimality for stochastic problems.

The rest of this paper is organized as follows. In Section II we propose a precise definition of the *principle of optimality* and show this is a necessary condition for Bellman's equation to be used for finding an optimal policy. Next, we consider a class of optimization problems called forward separable optimization problems. In Section III, we show that for any forward separable DP problem, there exists a separable augmented-state DP 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 define a class of functions, called naturally forward separable functions. We show DP problems with naturally forward separable objective functions can be tractable solved using state augmentation. In Section V we show how to approximate and numerically solve augmented-state dynamic programming problems. Furthermore, we extend our methodology to stochastic DP problems in Section VI and give a discretization scheme to solve stochastic DP's with additively separable objective functions in Section VII. We summarize how state augmentation can be used with discretization methods can be used to solve DP problems with non-separable objective functions in Section VIII. In Section IX we formulate and solve the battery scheduling problem as a non-separable DP problem.

II. Background: Dynamic Programming

In this paper, we propose a generalized class of dynamic programming problems. Specifically, we define the family of generalized dynamic programming problem as optimization problems of form $G(t_0,x_0)$, parameterized by $t_0 \in \mathbb{N}$ and $x_0 \in \mathbb{R}^n$, defined by an objective functions $J_{t_0,x_0} : \mathbb{R}^{m\times(T-t_0)} \times \mathbb{R}^{n\times(T-t_0+1)} \to \mathbb{R}$ 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,

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

$$\begin{aligned}
&\text{subject to:}\\
&x(t+1) = f(x(t),u(t),t) \text{ for } t = t_0,..,T\\
&x(t_0) = x_0,\ x(t) \in X_t \subset \mathbb{R}^n \text{ for } t = t_0,..,T\\
&u(t) \in U \subset \mathbb{R}^m \text{ for } t = t_0,..,T-1\\
&\mathbf{u} = (u(t_0),...,u(T-1)) \text{ and } \mathbf{x} = (x(t_0),...,x(T))
\end{aligned}$$

where $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{N} \to \mathbb{R}^n$ is measurable, $U \subset \mathbb{R}^m$ and $X_t \subset \mathbb{R}^n$ are bounded, $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ for all t. We denote $J^*_{t_0,x_0} = J_{t_0,x_0}(\mathbf{u}^*,\mathbf{x}^*)$.

We will call $\{x(t)\}_{t_0 \le t \le T}$ the state variables and $n = \dim(X_t)$ the state space dimension. Similarly we will call $\{u(t)\}_{t_0 \le t \le T-1}$ the input (control) variables and $m = \dim(U)$ the input (control) space dimension. For cases where the dimension of the state variable, $x(t)$, varies with time, we slightly abuse notation and define the state space dimension as $\max_{t_0 \le t \le T} \dim(X_t)$.

Definition 1. *The function $J_{t_0,x_0} : \mathbb{R}^{m\times(T-t_0)} \times \mathbb{R}^{n\times(T-t_0+1)}$ is said to be additively separable if there exists functions, $c_T(x) : \mathbb{R}^n \to \mathbb{R}$, and $c_t(x,u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for $t = t_0, \cdots T-1$ such that,*

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

where $\mathbf{u} = (u(t_0),...,u(T-1))$ and $\mathbf{x} = (x(t_0),...,x(T))$.

The average mapped magnitude of the state vector over the time interval, $J(\mathbf{u},\mathbf{x}) = \frac{1}{T}\sum_{t=0}^{T} a_t(x(t))$ where $a_t : \mathbb{R}^n \to \mathbb{R}$, is clearly an example of an additively separable function. However later on we will see that variance type functions (11) are not additively separable.

Definition 2. *We say the sequence of inputs $\mathbf{u} = (u(t_0),...,u(T-1)) \in \mathbb{R}^{m\times(T-t_0)}$ is feasible if $u(t) \in U$ for $t = t_0,..,T-1$ 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 $f(x,u,t) \in X_{t+1}$. 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 $\mathbf{u} = (u(t),....,u(T-1))$ is feasible with $\mathbf{x} = (x(t),\cdots,x(T))$ for $G(t,x(t))$ and $\mathbf{v} = (v(s),....,v(T-1))$ if feasible with $\mathbf{h} = (h(s),\cdots,h(T))$ for $G(s,x(s))$ where $s > t$, then $\mathbf{w} = (u(t),\cdots,u(s-1),v(s),....,v(T-1))$ with $\mathbf{z} = (x(t),\cdots,x(s-1),h(s),\cdots,h(T))$ is 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 3. *A (Markov) 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 denote the set of policies corresponding to some optimization problem as Π. We say that π^* is an optimal policy for Problem (1) if*

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

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

The "Principle of Optimality" defines a class of optimization problems that satisfy Bellman's equation (4) and from which an optimal policy can be retrieved.

Definition 4. *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 $t_0 \le t < s < T$, if $\mathbf{u}^* = (u(t),...,u(T-1))$ and $\mathbf{x}^* = (x(t),...,x(T))$ solve $G(t,x(t))$ then $\mathbf{v} = (u(s),...,u(T-1))$ and $\mathbf{h} = (x(s),...,x(T))$ solve $G(s,x(s))$.*

The classical form of the dynamic programming algorithm, as originally defined in [1], can be used to solve indexed optimization problems of the Form (1) with an additively separable objective function. We denote this class of optimization problems by $P(t_0,x_0)$:

$$\min_{\mathbf{u},\mathbf{x}} J_{t_0,x_0}(\mathbf{u},\mathbf{x}) = \sum_{t=t_0}^{T-1} c_t(x(t),u(t)) + c_T(x(T)) \qquad (3)$$

$$\begin{aligned}
&\text{subject to:}\\
&x(t+1) = f(x(t),u(t),t) \text{ for } t = t_0,..,T\\
&x(t_0) = x_0,\ x(t) \in X_t \subset \mathbb{R}^n \text{ for } t = t_0,..,T\\
&u(t) \in U \subset \mathbb{R}^m \text{ for } t = t_0,..,T-1\\
&\mathbf{u} = (u(t_0),...,u(T-1)) \text{ and } \mathbf{x} = (x(t_0),...,x(T)).
\end{aligned}$$

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

Lemma 5. *Any problem of Form $P(t_0,x_0)$ in (3) satisfies the principle of optimality.*

Proof. Suppose $\mathbf{u}^* = (u(t),...,u(T-1))$ and $\mathbf{x}^* = (x(t),...,x(T))$ solve $P(t,x(t))$ in (2). Now we suppose by contradiction that there exists some $s > t$ such that $\mathbf{v} = (u(s),...,u(T-1))$ and $\mathbf{h} = (x(s),...,x(T))$ 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 $J_{s,x(s)}(\mathbf{w},\mathbf{z}) < J_{s,x(s)}(\mathbf{v},\mathbf{h})$. 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 $\hat{\mathbf{u}} = (u(t),...,u(s-1),w(s),...,w(T-1))$ and $\hat{\mathbf{x}} = (x(t),...,x(s-1),z(s),...,z(T-1))$. 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. $\square$

Proposition 6 ([28])**.** *For optimization problems of the form $P(t,x)$ in (3) with optimal objective values $J^*_{t,x}$, define the function $F(x,t) = J^*_{t,x}$. Then the following hold for all $x \in X_t$,*

$$\begin{aligned}
&F(x,t) = \inf_{u\in\Gamma_{t,x}} \{c_t(x,u) + F(f(x,u,t),t+1)\}\ \forall t \in \{t_0,..,T-1\} \qquad (4)\\
&F(x,T) = c_T(x) \qquad \forall x \in X_T.
\end{aligned}$$

Equation (4) is often referred to as Bellman's equation and a function F which satisfies Bellman's equation is often referred to as the "optimal cost-to-go" function. Prop. 6 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. A solution to Bellman's equation provides a state-feedback law or optimal policy as follows.

Corollary 7 ([28]). *Consider $P(t_0,x_0)$ in (3). Suppose $F(x,t)$ satisfies Equation (4) for $P(t_0,x_0)$, then if there exists a policy such that,*

$$\theta(x,t) = \arg\min_{u\in\Gamma_{t,x}} \{c_t(x,u) + F(f(x,u,t),t+1)\},$$

then θ is a optimal policy for the problem $P(t_0,x_0)$.

Dynamic Programming with Maximum 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:

$$\begin{aligned}
&\min_{\mathbf{u},\mathbf{x}} J_{t_0,x_0}(\mathbf{u},\mathbf{x}) := \sum_{t=t_0}^{T-1} c_t(x(t),u(t)) + c_T(x(T)) + \max_{t_0\le k\le T} d_k(x(k))\\
&\text{subject to:} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (5)\\
&x(t+1) = f(x(t),u(t),t) \text{ for } t=t_0,..,T\\
&x(t_0)=x_0,\ x(t)\in X_t\subset\mathbb{R}^n \text{ for } t=t_0,..,T\\
&u(t)\in U\subset\mathbb{R}^m \text{ for } t=t_0,..,T-1\\
&\mathbf{u}=(u(t_0),...,u(T-1)) \text{ and } \mathbf{x}=(x(t_0),...,x(T)),
\end{aligned}$$

where $c_T(x):\mathbb{R}^n\to\mathbb{R}$; $c_t(x,u):\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ for $t_0\le t\le T-1$; $d_t(x):\mathbb{R}^n\to\mathbb{R}$ for $t=t_0,\cdots T$; $f:\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{N}\to\mathbb{R}^n$.

Counter Example 8. *The class of optimization problems of the form $S(t_0,x_0)$ in (5) does not satisfy the principle of optimality.*

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

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

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 $\mathbf{u}^*=(h,-h,h)$, yielding an optimal trajectory of $\mathbf{x}^*=\{0,h,0,h\}$. Following this input sequence until $t=2$ we examine the problem $S(2,0)$.

$$\begin{aligned}
&\min_{u(2)\in\mathbb{R},0\le x(3)\le h} c_2(u(2)) + \max_{2\le k\le 3} x(k)\\
&\text{subject to: } x(t+1)=x(t)+u(t),\\
&x(2)=0,\ 0\le x(t)\le h,\ u(t)\in\{-h,0,h\}
\end{aligned}$$

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

For this sub-problem, there are two feasible inputs: $u(2)\in\{0,h\}$. Of these, the first 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 $S(2,0)$. $\square$

III. Constructing Equivalent Additively Separable DP Problems To Forward-Separable DP Problems

In this section we will define the class of forward separable objective functions. 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.

Forward separable functions were first defined in [29]. Intuitively, this is the class of functions that can be separated into a composition of maps ordered forward in time. In the next definition we will build upon the concept of forward separability by introducing the notion of augmented dimension.

Definition 9. *The function $J:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$ is said to be forward separable if there exists representation maps $\phi_{t_0}:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^{d_{t_0}}$, $\phi_T:\mathbb{R}^n\times\mathbb{R}^{d_{T-1}}\to\mathbb{R}$, and $\phi_i:\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^{d_{i-1}}\to\mathbb{R}^{d_i}$ for $i=t_0+1,\cdots T-1$ such that*

$$\begin{aligned}
J(\mathbf{u},\mathbf{x}) = \phi_T(x(T),\phi_{T-1}(x(T-1),u(T-1),\phi_{T-2}(...., \qquad (6)\\
\phi_{t_0+1}(x(t_0+1),u(t_0+1),\phi_{t_0}(x(t_0),u(t_0)))),....,))),
\end{aligned}$$

where $\mathbf{u}=(u(t_0),...,u(T-1))\in\mathbb{R}^{m\times(T-t_0)}$ and $u(i)\in\mathbb{R}^m$ for $i\in\{t_0,...,T-1\}$; $\mathbf{x}=(x(t_0),...,x(T))\in\mathbb{R}^{n\times(T+1-t_0)}$ and $x(i)\in\mathbb{R}^n$ for $i\in\{t_0,...,T\}$; $d_i\in\mathbb{N}$ for $i\in\{t_0,...,T-1\}$.

Moreover we say $J(\mathbf{u},\mathbf{x})$ is forward separable and has a representation dimension of l if there exists $\{\phi_i\}$ that satisfies (6) and $l=\max_{i\in\{t_0,...,T-1\}}\{d_i\}$ where $d_i=\dim(Im\{\phi_i\})$.

Note: The representation dimension of a forward separable function is a property of the set $\{\phi_i\}$ chosen and not the function. The representation dimension of a forward separable function is not unique.

Clearly, any additively separable objective function of the form $J(\mathbf{u},\mathbf{x})=\sum_{t=t_0}^{T-1}c_t(u(t),x(t))+c_T(x(T))$ is forward separable and has a representation dimension of 1 using,

$$\begin{aligned}
&\phi_{t_0}(x,u) = c_{t_0}(x,u) \qquad\qquad (7)\\
&\phi_i(x,u,w) = c_i(x,u)+w \quad \text{for } i=t_0+1,\cdots,T-1\\
&\phi_T(x,w) = c_T(x)+w.
\end{aligned}$$

A. How State Augmentation Allows us to Solve Forward Separable Dynamic Programming Problems

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

$$\begin{aligned}
&\min_{\mathbf{u},\mathbf{x}} J_{t_0,x_0}(\mathbf{u},\mathbf{x}) && (8)\\
&\text{subject to:}\\
&x(t+1)=f(x(t),u(t),t) \text{ for } t=t_0,..,T\\
&x(t_0)=x_0,\ x(t)\in X_t\subset\mathbb{R}^n \text{ for } t=t_0,..,T\\
&u(t)\in U\subset\mathbb{R}^m \text{ for } t=t_0,..,T-1\\
&\mathbf{u}=(u(t_0),...,u(T-1)) \text{ and } \mathbf{x}=(x(t_0),...,x(T)),
\end{aligned}$$

where J_{t_0,x_0} is forward separable with associated ϕ_i. For any 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 $H(t_0,x_0)$ and which satisfies the principle of optimality. $A(t_0,x_0)$ is defined as follows,

$$\begin{aligned}
&\min_{\mathbf{u},\mathbf{z}} L_{t_0,z_0}(\mathbf{u},\mathbf{z})=z_2(T+1) && (9)\\
&\text{subject to:}\\
&\begin{bmatrix} z_1(t+1)\\ z_2(t+1)\end{bmatrix}=\begin{bmatrix} f(z_1(t),u(t),t)\\ \phi_t(z_1(t),u(t),z_2(t))\end{bmatrix} t_0\le t<T\\
&\begin{bmatrix} z_1(T+1)\\ z_2(T+1)\end{bmatrix}=\begin{bmatrix} f(z_1(T),u(T),T)\\ \phi_T(z_1(T),z_2(T))\end{bmatrix}\\
&\begin{bmatrix} z_1(t_0)\\ z_2(t_0)\end{bmatrix}=\begin{bmatrix} x_0\\ 0\end{bmatrix},\ z_1(t)\in X_t,\ u(t)\in U \text{ for } t=t_0+1,..,T\\
&\mathbf{u}=(u(t_0),...,u(T-1)) \text{ and } \mathbf{z}=\left(\begin{bmatrix} z_1(t_0)\\ z_2(t_0)\end{bmatrix},...,\begin{bmatrix} z_1(T)\\ z_2(T)\end{bmatrix}\right)
\end{aligned}$$

where $f:\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{N}\to\mathbb{R}^n$, $z_1(t)\in\mathbb{R}^n$, $z_2(t)\in\mathbb{R}^{d_t}$, $d_t=\dim(Im\{\phi_{t-1}\})$ and $u(t)\in\mathbb{R}^m$ for all t.

Lemma 10. *Suppose J_{t_0,x_0} is the objective function for the optimization problem $H(t_0,x_0)$ (8) and is forward separable with associated ϕ_i. Consider the augmented optimization problem $A(t_0,x_0)$ (9) and denote its objective function by L_{t_0,x_0}. Then $J^*_{t_0,x_0}=L^*_{t_0,x_0}$. Furthermore, suppose $\mathbf{u}$ and $\mathbf{x}=(x(t_0),...,x(T))$ solve $H(t_0,x_0)$ and $\mathbf{w}$ and $\mathbf{z}=\left(\begin{bmatrix} z_1(t_0)\\ z_2(t_0)\end{bmatrix},...,\begin{bmatrix} z_1(T)\\ z_2(T)\end{bmatrix}\right)$ 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 $\mathbf{z}_1:=(z_1(t_0),...,z_1(T))$ 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 $z_1(t_0)=x_0$ and $z_1(t+1)=f(z_1(t),u(t),t)$, we have $x(t)=z_1(t)\in X_t$ 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 $z_2(t+1)=\phi_t(z_1(t),u(t),z_2(t))$, $z_2(t_0+1)=\phi_0(z_1(t_0),u(t_0))$, $z_2(t_0)=0$, then $\mathbf{w}$ and $\mathbf{z}$ are feasible. Furthermore, since $H(t_0,x_0)$ has a forward separable objective function we have,

$$\begin{aligned}
J(\mathbf{u},\mathbf{x})=\ &\phi_T(z_1(T),\phi_{T-1}(z_1(T-1),w(T-1),\phi_{T-2}(....,\\
&\phi_{t_0+1}(z_1(t_0+1),w(t_0+1),\phi_{t_0}(z_1(t_0),w(t_0)))),....,))).
\end{aligned}$$

However, we now observe

$$\begin{aligned}
&z_2(T+1)=\phi_T(z_1(T),z_2(T))\\
&z_2(T)=\phi_{T-1}(z_1(T-1),u(T-1),z_2(T-1))\\
&\vdots\\
&z_2(t_0+1)=\phi_{t_0}(z_1(t_0),u(t_0)).\\
&z_2(t_0)=0.
\end{aligned}$$

Hence we have,

$$\begin{aligned}
L(\mathbf{w},\mathbf{z})&=z_2(T+1)\\
&=\phi_T(z_1(T),\phi_{T-1}[z_1(T-1),w(T-1),\phi_{T-2}\{....,\\
&\quad\phi_{t_0+1}\{z_1(t_0+1),w(t_0+1),\phi_{t_0}\{z_1(t_0),w(t_0)\}\}\},....,\}]).\\
&=J(\mathbf{u},\mathbf{x}).
\end{aligned}$$

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 $J^*_{t_0,x_0}$. ☐

Proposition 11. *The augmented optimization problem $A(t_0,x_0)$ in (9) satisfies the Principle of Optimality.*

Proof. $A(t_0,x_0)$ is a special case of $P(t_0,x_0)$ (3) where $c_i=0$ for $i\neq T$ and $c_T([z_1z_2]^T)=z_2$. Lemma 5 shows optimization problems of the form $P(t_0,x_0)$ satisfy the principle of optimality. ☐

Lemma 10 tells us that for any forward separable problem of the form $H(t_0,x_0)$ (8) there exists an equivalent optimization problem of the form $A(t_0,x_0)$ (9). Furthermore Proposition 11 shows that $A(t_0,x_0)$ satisfies the principle of optimality. Therefore a solution for $H(t_0,x_0)$ can be found by recursively solving Bellman's equation (4) for $A(t_0,x_0)$.

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 past information about the system. In this context, the augmented state contains the historic information necessary to make the correct decision at the present time. However by adding augmented states we increase the state space dimension and the complexity of the optimization problem.

Corollary 12. *Suppose the forward separable function, $J:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$, is the objective function for the optimization problem $H(t_0,x_0)$ (8) and has a representation dimension of l. Then the associated augmented optimization problem with this representation, $A(t_0,x_0)$ (9), has a state space of dimension $l+n$ and input space of dimension m.*

Proof. Follow as $z_1(t)\in\mathbb{R}^n$ and $z_2(t)\in\mathbb{R}^{d_t}$, where $d_t=\dim(Im\{\phi_{t-1}\})$ and the functions $\{\phi_t\}_{t_0}^T$ are the representation maps of J. As J has representation dimension l we have $d_t\le l$ for all $t_0\le t\le T$. Thus it follows the state space dimension is $n+\max_{t_0\le t\le T} d_t=n+l$. ☐

IV. When State Augmentation Can Be Used To Tractably Solve Forward Separable DP Problems

It is well known DP problems cannot be numerically tractably solved when their state space dimension is too high;

this is often called "the curse of dimensionality". As shown in Corollary 12, the state space dimension of a non-separable DP problem is smaller than that of the associated augmented optimization problem; this difference in dimension depends on the representation dimension of the forward separable objective function. In this section we define a class objective functions, called Naturally Forward Separable Functions (NFSF's), with representation dimension independent of the number of time steps and the dimension of the state and input space.

Next we will show that it is possible to represent any function as a forward separable function. To do this we introduce some notation. For a vector $v=(v_1,...,v_n)^T \in \mathbb{R}^n$ we define $[v]_i^j=(v_i,...,v_j)$ for some $1 \le i < j \le n$.

Lemma 13. *Any function $J:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$ can be shown to be forward separable with a representation of dimension $l(n,m,T-t_0)=(T-t_0)(n+m)$.*

Proof. Consider some function $J:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$. To show J is forward separable we will give functions $\{\phi_i\}_{i=t_0}^T$ that satisfy (6).

The function $\phi_{t_0}:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^{n+m}$ is defined by,

$$\phi_{t_0}(x,u)=[x^T,u^T]=\left[x_1,...,x_n,u_1,...,u_m\right].$$

For $i\in\{t_0+1,...T-1\}$ the function $\phi_i:\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^{(i-t_0)(n+m)}\to\mathbb{R}^{(i+1-t_0)(n+m)}$ is defined by,

$$\phi_i(x,u,w)=\left[[w]_1^{n(i-t_0)},x^T,[w]_{n(i-t_0)+1}^{(i-t_0)(n+m)},u^T\right].$$

The function $\phi_T:\mathbb{R}^n\times\mathbb{R}^{(T-t_0)(n+m)}\to\mathbb{R}$ is defined by,

$$\phi_T(x,w)=J([[w]_1^{n(T-t_0)},x],[w]_{n(T-t_0)+1}^{(n+m)(T-t_0)}).$$

Moreover it can be seen that the maximum dimension of the images of the maps $\{\phi_i\}_{i=t_0}^T$ is $(T-t_0)(n+m)$ showing the dimension of this representation of J is $l(n,m,T-t_0)=(T-t_0)(n+m)$. ☐

In the above approach to show that $J(\mathbf{u},\mathbf{x})$ is forward separable we naively took the strategy of using the functions $(\phi_i)_{t_0\le i\le T}$ to act like memory functions; that is to store the entire historic state trajectory and input sequence used. If $J(\mathbf{u},\mathbf{x})$ is the objective function for some optimization problem $H(t_0,x_0)$ (8) then this approach would result in the associated augmented optimization problem, $A(t_0,x_0)$ (9), having a very large state space dimension. Corollary 12 shows taking this naive approach results in the problem $A(t_0,x_0)$ having state space dimension $(T-t_0)(n+m)+n$. For a large number of time-steps, $T-t_0$, $A(t_0,x_0)$ will be intractable. For this reason we next define a special class of forward separable functions that have a representation with dimension independent of the number of time-steps.

Definition 14. *We say a function $J:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$ is a Naturally Forward Separable Function (NFSF) if there exists maps, $\{\phi_i\}_{i=t_0}^T$, that satisfy* (6) *with associated representation dimension independent of n, m and T.*

A. *Algebra Of Naturally Forward Separable Functions*

Given a function, $J:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$, there is no known algorithm that computes representation maps, $\{\phi_i\}_{i=t_0}^T$, that can verify J is a NFSF. However, in this section we show how known NFSF's can combine together to construct new, more complex, NFSF's. Therefore, one approach to finding representation maps for a function, J, is to use NFSF's with known representation maps as "building blocks" to construct J. Later on, in Section IV-B we will give several examples of representations of NFSF's that can be used as such "building blocks".

Lemma 15. *Consider the naturally forward separable functions, $J_1:\mathbb{R}^{m_1\times(T_1-t_1)}\times\mathbb{R}^{n_1\times(T_1+1-t_1)}\to\mathbb{R}$ and $J_2:\mathbb{R}^{m_2\times(T_2-t_2)}\times\mathbb{R}^{n_2\times(T_2+1-t_2)}\to\mathbb{R}$, with representation dimensions l_1 and l_2 respectively. The functions $G_1(\mathbf{u},\mathbf{x})=J_1(\mathbf{u},\mathbf{x})+J_2(\mathbf{u},\mathbf{x})$ and $G_2(\mathbf{u},\mathbf{x})=J_1(\mathbf{u},\mathbf{x})J_2(\mathbf{u},\mathbf{x})$ are both naturally forward separable functions with representation dimension less then or equal to l_1+l_2.*

Proof. For simplicity us consider the case $t_1=t_2$ and $T_1=T_2$; other cases follow by the same argument. As J_1 and J_2 are forward separable functions there exists $\{g_i\}$ and $\{h_i\}$ such that J_1 and J_2 can be written in the form (6) with associated representation dimensions l_1 and l_2 respectively. We now show that G_1 is forward separable by defining the functions $\{\phi_i\}$ such that G_1 can be written in the form (6).

$$\phi_{t_1}(x,u)=\begin{bmatrix}g_{t_1}(x,u)\\h_{t_1}(x,u)\end{bmatrix}, \tag{10}$$

$$\phi_i(x,u,w)=\begin{bmatrix}g_i(x,u,[w]_1^{d_{i-1}})\\h_i(x,u,[w]_{d_{i-1}+1}^{d_{i-1}+s_{i-1}})\end{bmatrix} \text{ for } i\in\{t_1+1,....,T_1-1\}$$

$$\phi_{T_1}(x,u,w)=g_T(x,u,[w]_1^{d_{T_1-1}})+h_T(x,u,[w]_{d_{T_1-1}+1}^{d_{T_1-1}+s_{T_1-1}}),$$

where $d_i=\dim(Im\{g_i\})$ and $s_i=\dim(Im\{h_i\})$ for $i\in\{t_1,...,T_1-1\}$.

We conclude that G_1 has a representation dimension, denoted l_{G_1}, such that

$$\begin{aligned}l_{G_1}&=\max_{i\in\{t_1,...,T_1-1\}}\{d_i+s_i\}\\&\le\max_{i\in\{t_0,...,T-1\}}\{d_i\}+\max_{i\in\{t_0,...,T-1\}}\{s_i\}\\&=l_1+l_2.\end{aligned}$$

Furthermore, by a similar argument it can be shown G_2 is forward separable with representation dimension less than or equal to l_1+l_2.

Since J_1 and J_2 are NFSF's l_1 and l_2 are independent of n_i, m_i and T_i for $i=1,2$. Therefore there exists representations of G_1 and G_2 with dimensions independent of n_i, m_i and T_i for $i=1,2$, making G_1 and G_2 both NFSF's. ☐

Lemma 16. *Consider the function $J:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$ such that,*

$$J(\mathbf{u},\mathbf{x})=U\left(G(\mathbf{u},\mathbf{x})\right)$$

where $\mathbf{u}=(u(t_0),...,u(T-1))$, $u(t)\in\mathbb{R}^m$, $\mathbf{x}=(x(t_0),...,x(T))$, $x(t)\in\mathbb{R}^n$, $G:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$ is a NFSF with *representation dimension l_G, and $U:\mathbb{R}\to\mathbb{R}$. Then J is a NFSF and has a representation dimension less than or equal to l_G.*

Proof. Follows trivially using G's representation maps. ☐

B. Examples Of Naturally Forward Separable Functions

The fist example of a NFSF we give is related to risk measures and certainty equivalents, as discussed in [12].

Example 17. *The function $J:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$ such that,*

$$J(\mathbf{u},\mathbf{x}) = U\left(\sum_{t=t_0}^{T-1} c_t(x(t),u(t))\right)$$

where $U:\mathbb{R}\to\mathbb{R}$, is a NFSF with representation dimension 1.

Proof. The function $\sum_{t=t_0}^{T-1} c_t(x(t),u(t))$ is a NFSF using the representation maps given in (7). It therefore follows J is a NFSF by Lemma 16. ☐

Example 18. *The p-norm function given by*

$$J(\mathbf{u},\mathbf{x}) = \left(\sum_{t=t_0}^{T-1} ||x(t)||_2^p\right)^{\frac{1}{p}},$$

where $||\cdot||_2$ is the euclidean norm and $p>0$, is a NFSF with representation dimension 1.

Proof. Follows by Example 17 using $U(x)=x^{\frac{1}{p}}$ and $c_t(x,u)=||x||_2^p$. ☐

We next give a NFSF that can be thought of as a discrete time version of the Green measure; when used as an objective function for a DP problem it measures the amount of time the state and input spend in some set.

Example 19. *Consider the function $J:\mathbb{R}^{m\times(T-t_0)}\times\mathbb{R}^{n\times(T+1-t_0)}\to\mathbb{R}$ such that,*

$$J(\mathbf{u},\mathbf{x}) = |\{i\in\{t_0,\dots,T\}:(x(i),u(i))\in S\}|$$

where $\mathbf{u}=(u(t_0),\dots,u(T-1))$, $u(t)\in\mathbb{R}^m$, $\mathbf{x}=(x(t_0),\dots,x(T))$, $x(t)\in\mathbb{R}^n$, $S\subset\mathbb{R}^n\times\mathbb{R}^m$ and for $B\subset\mathbb{N}$ we denote $|B|$ to be the cardinality of the set B. Then J is a NFSF and has a representation of dimension 1.

Proof. We present functions such that $J(\mathbf{u},\mathbf{x})$ can be written in the form (6).

The function $\phi_{t_0}:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ is defined by,

$$\phi_{t_0}(x,u) = \begin{cases} 1 & \text{if } (x,u)\in S \\ 0 & \text{otherwise} \end{cases}.$$

The function $\phi_t:\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}\to\mathbb{R}$ is defined by,

$$\phi_t(x,u,w) = \begin{cases} w+1 & \text{if } (x,u)\in S \\ w & \text{otherwise} \end{cases} \quad \text{for } 1\le t\le T-1.$$

The function $\phi_T:\mathbb{R}^n\times\mathbb{R}\to\mathbb{R}$ is defined by,

$$\phi_T(x,w) = \begin{cases} w+1 & \text{if } (x,u)\in S \\ w & \text{otherwise} \end{cases}.$$

Moreover it can be seen that the maximum dimension of the images of the maps $\{\phi_i\}_{i=t_0}^T$ is 1 showing the dimension of this representation of J is 1. ☐

Example 20. *Consider the variance type function, $J:\mathbb{R}^{m\times T}\times\mathbb{R}^{n\times(T+1)}\to\mathbb{R}$ defined by,*

$$J(\mathbf{u},\mathbf{x}) = \sum_{t=0}^{T}\left[a_t(x(t)) - \frac{1}{T}\sum_{s=0}^{T} a_s(x(s))\right]^2 \tag{11}$$

where $\mathbf{u}=(u(0),\dots,u(T-1))$, $u(t)\in\mathbb{R}^m$, $\mathbf{x}=(x(0),\dots,x(T))$, $x(t)\in\mathbb{R}^n$, $a:\mathbb{R}^n\to\mathbb{R}$. Then J is a NFSF and has a representation dimension of 2.

Proof. Expanding the right hand side of (11) as in [9] we get,

$$\begin{aligned} &J(\mathbf{u},\mathbf{x}) \\ &= \sum_{t=0}^{T}\left[a_t^2(x(t)) - \frac{2}{T}a_t(x(t))\sum_{s=0}^{T}a_s(x(s)) + \frac{1}{T^2}\left(\sum_{s=0}^{T}a_s(x(s))\right)^2\right] \\ &= \sum_{t=0}^{T}a_t^2(x(t)) - \frac{1}{T}\left[\sum_{s=0}^{T}a_s(x(s))\right]^2. \end{aligned}$$

We now present functions such that $J(\mathbf{u},\mathbf{x})$ can be written in the form (6). The function $\phi_{t_0}:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}^2$ is defined by,

$$\phi_0(x,u) = \begin{bmatrix} a_1^2(x) \\ a_1(x) \end{bmatrix}.$$

The function $\phi_i:\mathbb{R}^n\times\mathbb{R}^m\times\mathbb{R}^2\to\mathbb{R}^2$ is defined by,

$$\phi_i(x,u,[w_1,w_2]^T) = \begin{bmatrix} w_1+a_i^2(x) \\ w_2+a_i(x) \end{bmatrix} \quad \text{for } 1\le i\le T-1.$$

The function $\phi_T:\mathbb{R}^n\times\mathbb{R}^2\to\mathbb{R}$ is defined by,

$$\phi_T(x,[w_1,w_2]^T) = (w_1+a_T^2(x)) - \frac{1}{T}(w_2+a_T(x))^2.$$

Moreover it can be seen that the maximum dimension of the images of the maps $\{\phi_i\}_{i=t_0}^T$ is 2 showing the dimension of this representation of J is 2. ☐

We now show that the maximum function, that appears in the objective function of the battery scheduling problem in Section IX, is a NFSF.

Example 21. *Consider the function $J:\mathbb{R}^{m\times T}\times\mathbb{R}^{n\times(T+1)}\to\mathbb{R}$ such that,*

$$J(\mathbf{u},\mathbf{x}) = \max\{\max_{0\le k\le T-1}\{c_k(u(k),x(k))\},c_T(x(T))\}$$

where $\mathbf{u}=(u(0),\dots,u(T-1))$, $u(t)\in\mathbb{R}^m$, $\mathbf{x}=(x(0),\dots,x(T))$, $x(t)\in\mathbb{R}^n$, $c_k:\mathbb{R}^m\times\mathbb{R}^n\to\mathbb{R}$ for $0\le k\le T-1$ and $c_T:\mathbb{R}^n\to\mathbb{R}$. Then J is a NFSF and has a representation dimension of 1.

Proof.

$$\begin{aligned} J(\mathbf{u},\mathbf{x}) &= \max\{\max_{0\le k\le T-1}\{c_k(u(k),x(k))\},c_T(x(T))\} \\ &= \max\{c_T(x(T)),\max\{c_{T-1}(u(T-1),x(T-1)),\cdots \\ &\quad \max\{..,\max\{c_1(u(1),x(1)),\max\{c_0(u(0),x(0))\}\},..\}\}. \end{aligned}$$

It is now clear we can write J in the form (6) as follows. The function $\phi_{t_0}:\mathbb{R}^n\times\mathbb{R}^m\to\mathbb{R}$ is defined by,

$$\phi_{t_0}(x,u) = c_{t_0}(x,u).$$

The function $\phi_i : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R} \to \mathbb{R}$ is defined by,

$$\phi_i(x,u,w) = \max(c_i(x,u),w) \text{ for } t_0+1 \le i \le T-1.$$

The function $\phi_T : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is defined by,

$$\phi_T(x,w) = \max(c_T(x),w).$$

Moreover it can be seen that the maximum dimension of the images of the maps $\{\phi_i\}_{i=t_0}^{T}$ is 1 showing the dimension of this representation of J is 1. □

V. Numerically Solving Deterministic Additively Separable DP Problems

In Section III we showed that all forward separable problems of the form $H(t_0,x_0)$ have an equivalent optimization problem of the form $A(t_0,x_0)$. Problems of the form $A(t_0,x_0)$ are special cases of problems of the form $P(t_0,x_0)$. In this section we show how to numerically solve problems of the form $P(t_0,x_0)$.

For implementation, we use an approximation scheme that maps our class of DP problems to a much simpler class of DP problems with finite state and control spaces. It is known for dynamic programming problems with countable state and control spaces the infimum in Bellman's equation (4) is attained and the optimal cost to go function, $F(x,t)$, can be computed by enumeration. Similar numerical schemes with convergence proofs can be found in [16] [15].

A. Construction of Approximated Tractable Optimization Problems

Consider the optimization problem $P(t_0,x_0)$ (3) with compact state and control spaces of the form $X = [\underline{x},\bar{x}]^n$ and $U = [\underline{u},\bar{u}]^m$. For optimization problems of this form it is not generally possible to solve Bellman's Equation (4). We thus need to consider a sequence of "close" optimization problems with countable state and control spaces. We define a sequence of approximated optimization problems indexed by k and denoted by $P_k(t_0,x_0)$,

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

subject to:

$x(t+1) = argmin_{y\in X_k}\{||y - f(x(t),u(t),t)||_2\}$

$x(t_0) = x_0,\ x(t) \in X_k \subset \mathbb{R}^n,\ u(t) \in U_k \subset \mathbb{R}^m$ for $t = t_0,..,T$

$\mathbf{u} = (u(t_0),...,u(T-1))$ and $\mathbf{x} = (x(t_0),...,x(T))$

Where $X_k = \{x_1,...,x_k\}^n$ such that $\underline{x} = x_1 < x_2 < ... < x_k = \bar{x}$ and $||x_{i+1} - x_i||_2 = \frac{\bar{x}-\underline{x}}{k}$ for $1 \le i \le k-1$, and $U_k = \{u_1,...,u_k\}^m$ such that $\underline{u} = u_1 < u_2 < ... < u_k = \bar{u}$ and $||u_{i+1} - u_i||_2 = \frac{\bar{u}-\underline{u}}{k}$ for $1 \le i \le k-1$.

B. Constructing a Feasible Policy from the Solution of the Approximated Optimization Problem

By iteratively solving Bellman's equation (4) we can find an optimal solution to $P_k(t_0,x_0)$ which we denote as (x_k^*,u_k^*). Because the vector fields that define the underlying dynamics of $P(t_0,x_0)$ and $P_k(t_0,x_0)$ are different, the solution (x_k^*,u_k^*) is not necessarily feasible for $P(t_0,x_0)$. However using the optimal policy for $P_k(t_0,x_0)$, π_k^*, we can construct a feasible policy for $P(t_0,x_0)$ in the following way,

$$\theta_k(x,t) = \arg\min_{u\in\Gamma_{t,x}} ||\pi_k^*(\arg\min_{y\in X_k}\{||y-x||_2\},t) - u||_2 \in \Pi \tag{13}$$

where we recall $\Gamma_{t,x}$ is the set of feasible inputs such that if $u \in \Gamma_{t,x}$ then $u \in U$ and $f(x,u,t) \in X$ for the optimization problem $P(t_0,x_0)$ (3).

C. Convergence of our Constructed Policy

Suppose $\theta_k(x,t)$, from (13), is a feasible policy for $P(t_0,x_0)$ from the optimal policy of $P_k(t_0,x_0)$ using (13). Let $\mathbf{u}_k = (\theta_k(x_0,t),...,\theta_k(x_k(T-1),T-1))$ and $\mathbf{x}_k = (x_k(t_0),...,x_k(T))$ where $x_k(t_0) = x_0$, $x_k(t+1) = f(x_k(t),\theta_k(x_k(t),t),t)$ and f is the vector field from $P_k(t_0,x_0)$. If $P(t_0,x_0)$ satisfies assumptions (A1) to (A4) in [15] then it is known,

$$\lim_{k\to\infty} ||J_{x_0,t_0}(\mathbf{u}_k,\mathbf{x}_k) - J^*_{x_0,t_0}|| = 0, \tag{14}$$

where $J_{x_0,t_0}(\mathbf{u}_k,\mathbf{x}_k)$ is the resulting value objective function of $P(t_0,x_0)$ when the policy θ_k is used and $J^*_{x_0,t_0}$ is the optimal value of the objective function.

VI. Extension to Stochastic Models

In the deterministic case we saw it was necessary for a DP problem to satisfy the Principle Of Optimality (POP), found in Definition 4, in order for us to solve such a problem using Bellman's Equation. It was shown DP problems with non-separable objective functions due not satisfy the POP; this was the primary motivation for using state augmentation to construct equivalent DP problems that satisfy the POP. Therefore, before proceeding to use state augmentation on stochastic DP problems, we rigorously define the POP for stochastic DP problems and show stochastic DP problems with additively separable objective functions (also known as MDP's) satisfy such condition. To the authors knowledge no such definition exists in the literature.

Before we introduce stochastic DP problems we define a map from a chosen policy, initial condition and random inputs to the trajectory, $\mathbf{x}$, followed by the underlying dynamics of the problem; this will clarify which random variables the expectation in the objective function is respect to. For simplicity we only consider stochastic DP problems with Gaussian distributed random variables, however our methods trivially generalize to any probability distribution.

Definition 22. *For a vector field $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{N} \times \mathbb{R}^q \to \mathbb{R}^n$, a set of optimal polices Π associated with some optimization problem, a starting time $t_0 \in \mathbb{N}$, and terminal time $T \in \mathbb{N}$, let us denote **the state map** by $\psi_{f,t_0} : \Pi \times \mathbb{R}^n \times \mathbb{N} \times \mathbb{R}^{q\times(T-t_0)} \to \mathbb{R}^n$. We say that $x = \psi_{f,t_0}(\pi,x_0,T,[\mathbf{v}]_{t_0}^{T-1})$ if $x = x(T)$ where $x(T)$ is a solution to the following recursion equations $x(t_0) = x_0$, $x(t+1) = f(x(t),\pi(x(t),t),t,v(t))$ for $t \in \{t_0,...,T-1\}$ and $[\mathbf{v}]_{t_0}^{T-1} = [v(t_0),..,v(T-1)] \in \mathbb{R}^{q\times(T-t_0)}$. We denote the image of the state vector under a set of instantiations $Y \subset \mathbb{R}^{q\times(T-t_0)}$ by $\psi_{f,t_0}(\pi,x_0,T,Y) = \{\psi_{f,t_0}(\pi,x_0,T,[\mathbf{v}]_{t_0}^{T-1}) \in \mathbb{R}^n : [\mathbf{v}]_{t_0}^{T-1} \in Y\}$.*

*We also denote **the trajectory map** by $\Phi_{f,t_0} : \Pi \times \mathbb{R}^n \times \mathbb{N} \times \mathbb{R}^{q\times(T-t_0)} \to \mathbb{R}^{m\times(T-t_0)} \times \mathbb{R}^{n\times(T-t_0+1)}$. We say that $(\mathbf{u},\mathbf{x}) = \Phi_{f,t_0}(\pi,x_0,T,[\mathbf{v}]_{t_0}^{T-1})$ if $\mathbf{u} = (\pi(x(t_0),t_0),...,\pi(x(T-1),T-1))$, and $\mathbf{x} = (x(t_0),...,x(T))$ is such that $x(t) = \psi_{f,t_0}(\pi,x_0,t,[\mathbf{v}]_{t_0}^{T-1})$ for $t \in \{t_0,...,T-1\}$.*

We define the class of general stochastic dynamic programming problems with forward separable objective as $H_s(t_0,x_0)$,

$$\pi^{H_s*} = \arg\min_{\pi\in\Pi} \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}} \left(J_{t_0,x_0}^{H_s}(\Phi_{f,t_0}(\pi,x_0,T,[\mathbf{v}]_{t_0}^{T-1}))\right) \tag{15}$$
$$\text{subject to: } \psi_{f,t_0}(\pi,x_0,t,[\mathbf{v}]_{t_0}^{t-1}) \in X_t \text{ for } t = t_0,..,T$$
$$\pi(x,t) \in U_t \text{ and } v(t) \in \mathbb{R}^q \sim \mathcal{N}(\mathbf{0},I_{q\times q}) \forall x \in X_t, \forall t = t_0,..,T-1,$$

where $J_{t_0,x_0}^{H_s} : \mathbb{R}^{m\times(T-t_0)} \times \mathbb{R}^{n\times(T-t_0+1)} \to \mathbb{R}$ is a forward separable function with associated representation $\{\phi_i\}_{i=t_0}^T$; $f : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{N} \times \mathbb{R}^q \to \mathbb{R}^n$; ψ_{f,t_0} and Φ_{f,t_0} are the state and trajectory map respectively defined in Definition 22; U_i is assumed to be some compact subset of $\mathbb{R}^{m\times(i-t_0)}$; $X_i \subset \mathbb{R}^{n\times(i-t_0+1)}$; $[\mathbf{v}]_{t_0}^{T-1} = [v(t_0),..,v(t-1)] \in \mathbb{R}^{q\times(T-t_0)}$; $\mathbb{E}_\mathbf{v}$ is the expectation with respect to the random variable $\mathbf{v}$. Define $J_{t_0,x_0}^{H_s*} = \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J_{t_0,x_0}^{H_s}(\Phi_{f,t_0}(\pi^{H_s*},x_0,T,[\mathbf{v}]_{t_0}^{T-1}))\right)$ as the expected cost of the optimal policy when applied to $H_s(t_0,x_0)$.

Change in Notation for Stochastic Problems: Unlike in the deterministic case the solution to stochastic dynamic programming problems, such as (15), is now a policy $\pi \in \Pi$ and not a definite input and state sequence $\mathbf{u}^* \in \mathbb{R}^{m\times(T-t_0)}$ and $\mathbf{x}^* \in \mathbb{R}^{n\times(T-t_0+1)}$, such as in (1). This is because the optimal sequence of inputs, $\mathbf{u}^*$, that results in an optimal trajectory, $\mathbf{x}^*$, will depend on the instantiation of the random variables. This change of notation demonstrates that the solution to dynamic programming problem involving stochastic dynamics no longer belongs to some finite dimensional space, $(\mathbf{u}^*,\mathbf{x}^*) \in \mathbb{R}^{m\times(T-t_0)} \times \mathbb{R}^{n\times(T-t_0+1)}$, but rather an infinite dimensional functional space $\pi^* \in \Pi$.

A. Stochastic Additively Separable DP Problems

In the special case when the objective function of (15) is an additively separable function, as per Definition 1, given as

$$J_{t_0,x_0}^{Q}(\mathbf{u},\mathbf{x}) = \sum_{t=t_0}^{T-1} c_t(x(t),u(t)) + c_T(x(T)), \tag{16}$$

we denote the optimization problem (15) by $Q(t_0,x_0)$.

Throughout the literature the optimization problem of form $Q(t_0,x_0)$ is related to a tuple, called a Markov Decision Process (MDP), of form $\{\{X_t\}_{t_0}^T, \{U_t\}_{t_0}^{T-1}, \psi, \{P_t\}_{t_0}^{T-1}, \{c\}_{t_0}^T\}$; where $\psi(x,v,t) = \{u \in U_t : f(x,u,t,v) \in X_{t+1}\}$, $Q_t(B|x,u) = \int_B \mathbb{1}_B(f(x,u,t,v))\phi(v)dv$, and $\phi(v)$ is the probability density function of the random variable v.

B. The Principle of Optimality for Stochastic Problems

As discussed in [3] the extension of the principle of optimality to the stochastic case is non-trivial. We next give an example from [30] of a stochastic dynamic programming problem which shows that an optimal policy may not be optimal for every instantiation of the random variables at future time steps.

Let us consider the following stochastic dynamic programming problem $W(0,x_0)$,

$$\pi^* = \arg\min_{\pi\in\Pi} \mathbb{E}_{v(0)}\left(J_{0,x_0}(\Phi_{f,0}(\pi,x_0,1,[v(0)]))\right) \tag{17}$$
$$\text{subject to: } v(0) \sim U[0,1], \quad x(0) = x_0.$$

Here $J_{t,x_0}(\mathbf{u},\mathbf{x}) = -\sum_{n=t}^{1} u(n)$, $f(x,u,t,v) = v$, and $\pi \in \Pi \iff \pi(x,t) \in \{0,1\} \forall x \in \mathbb{R}, t = 0,1$.

Counter Example 23. *The policy* $\pi(x,t) = \begin{cases} 1 \text{ if } x \in [0,1) \\ 0 \text{ if } x = 1 \end{cases}$ *is optimal for the problem $W(0,0)$ (17) but not optimal for the problem $W(1,1)$.*

Proof. Clearly $J_{0,0}(\mathbf{u},\mathbf{x}) \geq -2$ for all $(\mathbf{u},\mathbf{x}) \in \{0,1\}^2 \times \mathbb{R}^3$ and $J_{0,0}(\mathbf{u},\mathbf{x}) = -2$ attainable using the input $(u(0),u(1)) = (1,1)$; therefore any solution of $W(0,0)$ will minimize the objective function to a value of -2. Now using the law of total expectation we get,

$$\begin{aligned}
&\mathbb{E}_{v(0)}\left(J_{0,0}(\Psi_{f,0,1}(\pi,0,[v(0),v(1)]))\right) \\
&= -\mathbb{E}_{v(0)}\left(\pi(0,0) + \pi(v(0),1)\right) \\
&= -\pi(0,0) - \mathbb{E}_{v(0)}\left(\pi(v(0),1)|v(0) \in [0,1)\right)\mathbb{P}_{v(0)}(v(0) \in [0,1)) \\
&\quad - \mathbb{E}_{v(0)}\left(\pi(v(0),1)|v(0) = 0\right)\mathbb{P}_{v(0)}(v(0) = 0) \\
&= -2,
\end{aligned}$$

since the probability of a continuous random variable (such as a uniformly distributed random variable) taking a particular value is 0. Thus it follows the policy π is optimal for $W(0,0)$. Trivially π is not optimal for $W(1,1)$ as the value of the objective functions becomes 0 under π whereas the input $u(1) = 1$ produces a smaller objective function value of -1. $\square$

Clearly, for the stochastic DP problems of form $H_s(t_0,x_0)$ (15), such as $W(0,0)$ (17), the optimal policy π^* does not always result in the same trajectory $\mathbf{x} = (x(t_0),...,x(T))$ being followed; as this is dependent on the instantiations of the underlying random variables, $[\mathbf{v}]_{t_0}^{T-1}$. As Counter Example 23 has shown, there exits stochastic DP problems, with additively separable objective functions, that have optimal policies that are no longer optimal for future timesteps if certain instantiations of the underlying random variables are realized. Therefore, it is too restrictive to extend Definition 4, the principle of optimality for the deterministic case, to the stochastic case by requiring stochastic problems satisfying the principle of optimality to be such that their optimal policy is also optimal for each instantiation at any future time step. With this in mind and motivated by the work of [4] we now give a probabilistic definition of the principle of optimality for stochastic dynamic programming problems.

Definition 24. *For an optimization problem $H_s(t_0,x_0)$ with optimal policy $\pi^* \in \Pi$ and associated state map ψ_{f,t_0}, defined in definition 22, let us denote the set indexed by $k \geq t_0$,*

$$\begin{aligned}
Y_k = \{&[\mathbf{v}]_{t_0}^{k-1} \in \mathbb{R}^{q\times(k-t_0)} : \\
&\pi^* \text{ does not solve } H_s(k,\psi_{f,t_0}(\pi^*,x_0,k,[\mathbf{v}]_{t_0}^{k-1}))\}
\end{aligned}$$

where $[\mathbf{v}]_{t_0}^{k-1} = [v(t_0),...,v(k-1)] \in \mathbb{R}^{q\times(k-t_0)}$. *We say stochastic optimization problems of the form* $H_s(t_0,x_0)$ (15) ***satisfy the principle of optimality*** *if for any* $k \geq t_0$ *we have*

$$\mathbb{P}_{[\mathbf{v}]_{t_0}^{k-1}}([\mathbf{v}]_{t_0}^{k-1} \in Y_k) = 0.$$

Where $\mathbb{P}_{[\mathbf{v}]_{t_0}^{k-1}}$ *is the probability measure associated with the random variable* $[\mathbf{v}]_{t_0}^{k-1} \in \mathbb{R}^{q\times(k-t_0)}$, $v(t) \sim \mathcal{N}(\mathbf{0}, I_{q\times q})$ *for* $t \in \{t_0,...,k-1\}$.

We next show that stochastic DP problems with additively separable objective functions (MDP's) must satisfy the principle of optimality, defined in Definition 24.

Lemma 25. *A problem of Form* $Q(t_0,x_0)$ (16) *satisfies the Principle of Optimality defined in Definition 24.*

Proof. Suppose π^* solves $Q(t_0,x_0)$. For $k > t_0$ and the state map ψ_{f,t_0} associated with $Q(t_0,x_0)$ let us recall the set defined in Definition 24,

$$Y_k := \{[\mathbf{v}]_{t_0}^{k-1} \in \mathbb{R}^{q\times(k-t_0)} : \pi^* \text{ does not solve } Q(k, x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1}))\}.$$

Where $[\mathbf{v}]_{t_0}^{k-1} := [v(t_0),...,v(k-1)] \in \mathbb{R}^{q\times(k-t_0)}$, and we use the short-hand $x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1}) := \psi_{f,t_0}(\pi^*, x_0, k, [\mathbf{v}]_{t_0}^{k-1})$.

Now for contradiction suppose there exists $k \in \{t_0,...,T\}$ such that $\mathbb{P}_{[\mathbf{v}]_{t_0}^{k-1}}([\mathbf{v}]_{t_0}^{k-1} \in Y_k) > 0$; where $v(t) \sim \mathcal{N}(\mathbf{0}, I_{q\times q})$ for $t \in \{t_0,...,k-1\}$. For $[\mathbf{v}]_{t_0}^{k-1} \in Y_k$ we know the policy π^* is not optimal for $Q(k, x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1}))$ and thus there exists a feasible policy $\theta \in \Pi$ such that,

$$\mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{k,x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1})}(\Phi_{f,k}(\theta, x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1}), T, [\mathbf{v}]_k^{T-1}))\middle|[\mathbf{v}]_{t_0}^{k-1} \in Y_k\right) \tag{18}$$
$$< \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{k,x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1})}(\Phi_{f,k}(\pi^*, x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1}), T, [\mathbf{v}]_k^{T-1}))\middle|[\mathbf{v}]_{t_0}^{k-1} \in Y_k\right).$$

Now let us consider the map,

$$\hat{\pi}_t([x(t_0),...,x(t)]) = \begin{cases} \theta(x(t),t) \text{ if } t \geq k, x(k) \in \psi_{f,t_0}(\pi^*, x_0, k, Y_k) \\ \pi^*(x(t),t) \text{ otherwise }. \end{cases} \tag{19}$$

Using Lemma 29 there exists a policy $\alpha \in \Pi$ such that (36) holds for $\{\hat{\pi}_t\}$ defined in (19). We will now show α contradicts that π^* is the optimal policy for $Q(t_0,x_0)$. We first note using (36) and the the law of total probabilities,

$$\begin{aligned} &\mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\alpha, x_0, T, [\mathbf{v}]_{t_0}^{T-1}))\right) \\ &= \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\hat{\pi}, x_0, T, [\mathbf{v}]_{t_0}^{T-1}))\right) \\ &= \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\hat{\pi}, x_0, T, [\mathbf{v}]_{t_0}^{T-1}))|[\mathbf{v}]_{t_0}^{k-1} \in Y_k\right) \\ &\qquad \mathbb{P}_{[\mathbf{v}]_{t_0}^{k-1}}([\mathbf{v}]_{t_0}^{k-1} \in Y_k) \\ &\quad + \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\hat{\pi}, x_0, T, [\mathbf{v}]_{t_0}^{T-1}))|[\mathbf{v}]_{t_0}^{k-1} \notin Y_k\right) \\ &\qquad \mathbb{P}_{[\mathbf{v}]_{t_0}^{k-1}}([\mathbf{v}]_{t_0}^{k-1} \notin Y_k). \end{aligned} \tag{20}$$

We recall the additive structure of $J^Q_{t_0,x_0}$

$$J^Q_{t_0,x_0}(\mathbf{u},\mathbf{x}) = \sum_{t=t_0}^{T-1} c_t(x(t),u(t)) + c_T(x(T)),$$

where $\mathbf{u} = (u(t_0),...,u(T-1))$ and $\mathbf{x} = (x(t_0),...,x(T))$ and $c_T(x) : \mathbb{R}^n \to \mathbb{R}$, $c_t(x,u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ for $t = t_0, \cdots T-1$.

Now using the fact $\hat{\pi}_t([x(t_0),...,x(t)]) = \pi^*(x(t),t)$ for all $t < k$, $\hat{\pi}_t([x(t_0),...,x(t)]) = \theta(x(t),t)$ if $t \geq k$ and $x(k) \in \psi_{f,t_0}(\pi^*, x_0, k, Y_k)$, linearity of the expectation and the inequality (18) we have,

$$\begin{aligned} &\mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,k}(\pi^*, x_0, T, [\mathbf{v}]_{t_0}^{T-1}))\middle|[\mathbf{v}]_{t_0}^{k-1} \in Y_k\right) \\ &= \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(\sum_{t=t_0}^{k-1} c_t(x_{\pi^*}([\mathbf{v}]_{t_0}^{t-1}), \pi^*(x_{\pi^*}([\mathbf{v}]_{t_0}^{t-1}),t))\middle|[\mathbf{v}]_{t_0}^{k-1} \in Y_k\right) \\ &+ \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{k,x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1})}(\Phi_{f,k}(\theta, x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1}), T, [\mathbf{v}]_k^{T-1}))\middle|[\mathbf{v}]_{t_0}^{k-1} \in Y_k\right) \\ &< \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(\sum_{t=t_0}^{k-1} c_t(x_{\pi^*}([\mathbf{v}]_{t_0}^{t-1}), \pi^*(x_{\pi^*}([\mathbf{v}]_{t_0}^{t-1}),t))\middle|[\mathbf{v}]_{t_0}^{k-1} \in Y_k\right) \\ &+ \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{k,x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1})}(\Phi_{f,k}(\pi^*, x_{\pi^*}([\mathbf{v}]_{t_0}^{k-1}), T, [\mathbf{v}]_k^{T-1}))\middle|[\mathbf{v}]_{t_0}^{k-1} \in Y_k\right) \\ &= \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\pi^*, x_0, T, [\mathbf{v}]_{t_0}^{T}))\middle|[\mathbf{v}]_{t_0}^{k-1} \in Y_k\right). \end{aligned} \tag{21}$$

Therefore using (20); the fact $\hat{\pi}_t([x(t_0),...,x(t)]) = \pi^*(x(t),t)$ if $x(k) \notin \psi_{f,t_0}(\pi^*, x_0, k, Y_{k,\pi^*})$; the total law of probability; the above inequality (21); and the assumption $\mathbb{P}_{[\mathbf{v}]_{t_0}^{k-1}}([\mathbf{v}]_{t_0}^{k-1} \in Y_k) > 0$ (so the inequality remains strict) we derive,

$$\begin{aligned} &\mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\alpha, x_0, T, [\mathbf{v}]_{t_0}^{T-1}))\right) \\ &\qquad < \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\pi^*, x_0, T, [\mathbf{v}]_{t_0}^{T-1}))\right). \end{aligned} \tag{22}$$

This contradicts the fact π^* is the optimal policy for $Q(t,x)$. Therefore we conclude $P_{[\mathbf{v}]_{t_0}^{k-1}}([\mathbf{v}]_{t_0}^{k-1} \in Y_k) = 0$ showing problems of the form $Q(t_0,x_0)$ satisfy Definition 24 and hence satisfy the principle of optimality. ☐

We will now state Bellman's equation for optimization problems of the form $Q(t,x)$.

Proposition 26 ([28]). *For optimization problems of the form* $Q(t,x)$ *in* (16) *with optimal objective values* $J^{Q*}_{t,x}$, *define the function* $F(x,t) = J^{Q*}_{t,x}$. *Then the following hold for all* $x \in X_t$,

$$F(x,t) = \inf_u \{c_t(x,u) + \mathbb{E}_v[F(f(x,u,t,v),t+1)]\}. \tag{23}$$
$$F(x,T) = c_T(x).$$

Furthermore we see in the next Corollary that if we are able to solve the stochastic Bellman equation (23) then we are able to construct the optimal policy that solves (16).

Corollary 27 ([28]). *Consider an optimization problem of the form* $Q(t_0,x_0)$ *in* (16). *Suppose* $F(x,t)$ *satisfies Equation* (23) *and suppose there exists a policy such that,*

$$\theta(x,t) = \arg\min_{u \in \Gamma_{t,x}} \{c_t(x,u) + \mathbb{E}_{\mathbf{v}}[F(f(x,u,t,v),t+1)]\}.$$

Then the policy θ *is a solution of* $Q(t_0,x_0)$.

C. *State Augmentation For Stochastic DP problems*

Analogous to the deterministic case shown in Lemma 10, for an optimization problem of the form $H_s(t_0,x_0)$ (15) we can use the separable representation $\{\phi_i\}_{i=t_0}^T$ of the objective function $J^{H_s}_{t_0,x_0}$ to construct an equivalent optimization problem fo form $Q(t_0,x_0)$ (16).

VII. Numerically Solving Additively Separable Stochastic DP Problems

In this section we show how to approximately solve $Q(t_0,x_0)$ (16), as it was shown in the previous section that for a DP, of form $H_s(t_0,x_0)$ (15), there exists an equivalent DP problem, of $Q(t_0,x_0)$.

To approximately solve $Q(t_0,x_0)$ we use a similar discretization scheme to the deterministic case detailed in Section V. However, unlike in the deterministic case, underlying random variables, $v \sim \mathcal{N}(0,1)$, possibly induce a non compact state space. Therefore before discretizing the state space we must first construct an approximate compact state space.

A. Constructing an Approximated Dynamic Programming Problem with Compact State Space

Consider the optimization problem $Q(t_0,x_0)$ with compact control space $U=[\underline{u},\bar{u}]^m$ and underlying random variables $v \sim \mathcal{N}(\mathbf{0},I_{q\times q})$. As in [15] we assume $\forall \varepsilon>0$ there exists a compact set $H_{\varepsilon,t}=[\underline{x}_{\varepsilon,t},\bar{x}_{\varepsilon,t}]^n \subset X$ (that depends on ε and t) such that $x_0 \in H_{\varepsilon,0}$ and,

$$\sup_{x\in H_{\varepsilon,t}, u\in U} \mathbb{P}_v(f(x,u,t,v)\notin H_{\varepsilon,t+1}) < \varepsilon. \tag{24}$$

We then construct the associated compact optimization problem to $Q(t_0,x_0)$ denoted by $Q_{\varepsilon,k}(t_0,x_0)$,

$$\begin{aligned} &\arg\min_{\pi\in\Pi}\ \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{\tilde{f},t_0}(\pi,x_0,T,[\mathbf{v}]_{t_0}^{T-1}))\right) \qquad (25)\\ &\text{subject to: } \ \psi_{\tilde{f},t_0}(\pi,x_0,t,[\mathbf{v}]_{t_0}^{t-1}) \in \tilde{X}_{\varepsilon,t,k} \text{ for } t=t_0,..,T\\ &\pi(x,t)\in\tilde{U}_k \text{ and } v(t)\in\mathbb{R}^q \sim \mathcal{N}(\mathbf{0},I_{q\times q})\forall x\in X_t, \forall t=t_0,..,T-1,\end{aligned}$$

where $\tilde{f}(x,u,t,v) = \arg\min_{y\in X_{\varepsilon,t+1,k}}\{||y-f(x,u,t,v)||_2\}$, $\tilde{X}_{\varepsilon,t,k}=\{x_{1,t},...,x_{k,t}\}^n$ such that $\underline{x}_{\varepsilon,t}=x_{1,t}<x_{2,t}<...<x_{k,t}=\bar{x}_{\varepsilon,t}$ and $||x_{i+1,t}-x_{i,t}||_2=\frac{\bar{x}_{\varepsilon,t}-\underline{x}_{\varepsilon,t}}{k}$ for $1\le i\le k-1$, $\tilde{U}_k=\{u_1,...,u_k\}^m$ such that $\underline{u}=u_1<u_2<...<u_k=\bar{u}$ and $||u_{i+1}-u_i||_2=\frac{\bar{u}-\underline{u}}{k}$ for $1\le i\le k-1$, and $[\mathbf{v}]_{t_0}^{T-1}=[v(t_0),..,v(T-1)]\in\mathbb{R}^{q\times(T-t_0)}$.

Analogous to the deterministic case the optimal policy $\pi^*_{\varepsilon,k}$ for $Q_{\varepsilon,k}(t_0,x_0)$ can be solved exactly by iteratively solving Bellman's equation (23). One can then construct a feasible policy for $Q(t_0,x_0)$ using,

$$\theta_{\varepsilon,k}(x,t)=\arg\min_{u\in\Gamma_{t,x}}||\pi^*_{\varepsilon,k}(\arg\min_{y\in X_{\varepsilon,t,k}}\{||y-x||_2\},t)-u||_2\in\Pi \tag{26}$$

where $\Gamma_{t,x}$ is the set of feasible controls at time $t\in\{0,...,T-1\}$ and state position $x\in\mathbb{R}^n$ for $Q(t_0,x_0)$ (16) and $X_{\varepsilon,t,k}$ is the state grid constraint in the problem $Q_{\varepsilon,k}(t_0,x_0)$ (25).

If $Q(t_0,x_0)$ satisfies assumption (A1) to (A4) in [15] then

$$\lim_{\varepsilon\to 0,k\to\infty}\left|\mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{\tilde{f},t_0}(\theta_{\varepsilon,k},x_0,T,[\mathbf{v}]_{t_0}^{T-1}))\right)-J^{Q*}_{t_0,x_0}\right|=0, \tag{27}$$

where $J^{Q*}_{t_0,x_0}=\mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\pi^{Q*},x_0,T,[\mathbf{v}]_{t_0}^{T-1}))\right)$ is the expected cost of the optimal policy when applied to $Q(t_0,x_0)$.

VIII. Solving Naturally Forward Separable DP Problems Using Augmentation And Discretization

Given a DP problem with a naturally forward separable objective function, with known representation maps, we have shown in Section III and Section VI how to construct equivalent DP problems with additively separable objective functions. We have furthermore proposed discretization schemes in Section V, for the deterministic case, and Section VII-A, for the stochastic case, to solve DP problems with additively separable objective functions. We now summarize the approach of augmenting and discretizing by proposing the following steps for solving a non-separable DP problem. Given a DP problem of the from $H(t_0,x_0)$ (8), or $H_s(t_0,x_0)$ (15) if stochastic, we do the following:

1) Find representation maps of the objective function to write it in the form (6). One approach to this is to use Section IV-A that details how to use NFSF's, with known representation maps, to find unknown representation maps of other NFSF's.
2) Construct the associated augmented optimization problem of form $P(t_0,x_0)$ (3), if deterministic, or $Q(t_0,x_0)$ (16), if stochastic.
3) Use discretization to approximate the augmented optimization problem to numerically tractable DP problems of Form $P_k(t_0,x_0)$ (12), if deterministic, or $Q_{\varepsilon,k}(t_0,x_0)$ (25), if stochastic.
4) Numerically solve $P_k(t_0,x_0)$ or $Q_{\varepsilon,k}(t_0,x_0)$ for sufficiently large discretization parameters, $k\in\mathbb{N}$ and $\varepsilon>0$.
5) Construct a feasible policy for the original DP problem from the optimal policy of $P_k(t_0,x_0)$ or $Q_{\varepsilon,k}(t_0,x_0)$ using (13) or (26).

To illustrate how we use state augmentation and discretization methods we consider the following DP problem from [8].

$$\begin{aligned} \min J = x(3)^2[u(0)^2+u(1)^2+u(1)u(2)^2]^{\frac{1}{2}} \qquad (28)\\ +[u(0)^2+u(1)^2+u(1)u(2)^2]^2\\ \text{subject to, }\quad x(t+1)=\frac{x(t)}{u(t)} \quad \text{for } t\in\{1,2,3\}\\ x(0)=10,\quad u(0),u(1),u(2)\ge 0.\end{aligned}$$

In [8] an analytic solution for (28) was found to be:

$$\mathbf{x}^*=\begin{bmatrix}10\\6.3943938\\5.782475\\3.8882658\end{bmatrix},\ \mathbf{u}^*=\begin{bmatrix}1.5638699\\1.105823\\1.4871604\end{bmatrix},\ J^*=74.767439.$$

The objective function J in (28) is a NFSF and has a representation dimension of 2. This can be shown by writing J in the form (6) using the functions,

$$\phi_0(x,u)=\begin{bmatrix}u^2\\0\end{bmatrix},\quad \phi_1\left(x,u,\begin{bmatrix}w_1\\w_2\end{bmatrix}\right)=\begin{bmatrix}w_1+u^2\\u\end{bmatrix}$$
$$\phi_2\left(x,u,\begin{bmatrix}w_1\\w_2\end{bmatrix}\right)=\begin{bmatrix}w_1+w_2^2u^2\\0\end{bmatrix},$$
$$\phi_3\left(x,\begin{bmatrix}w_1\\w_2\end{bmatrix}\right)=\begin{bmatrix}x^2\sqrt{w_1}+w_1^2\\0\end{bmatrix}.$$

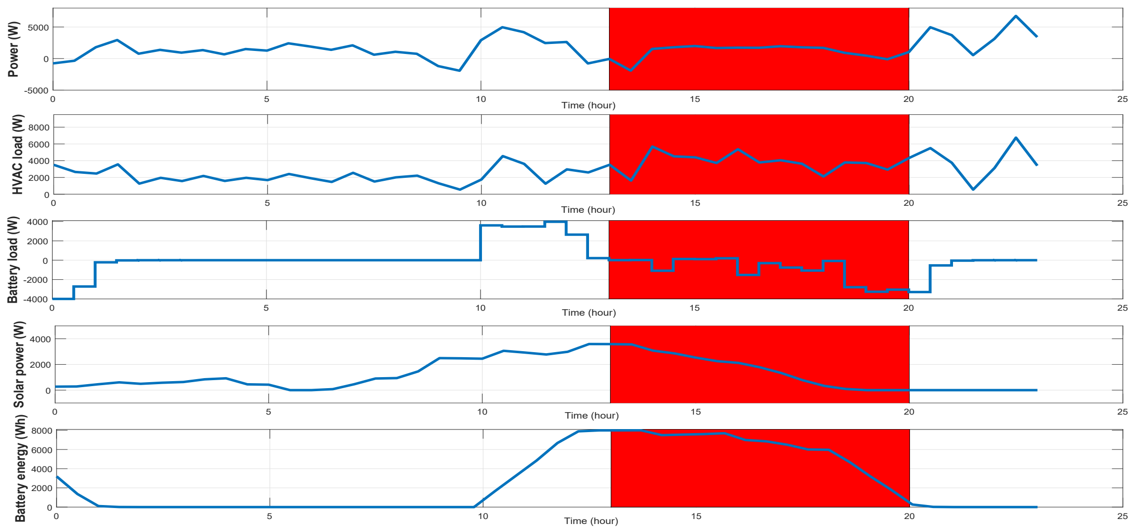

Figure 1. The trajectory the algorithm produces for randomly generated stochastic solar data. The supremun of the power is 1.05788(kw) and the cost is \$47.7211.

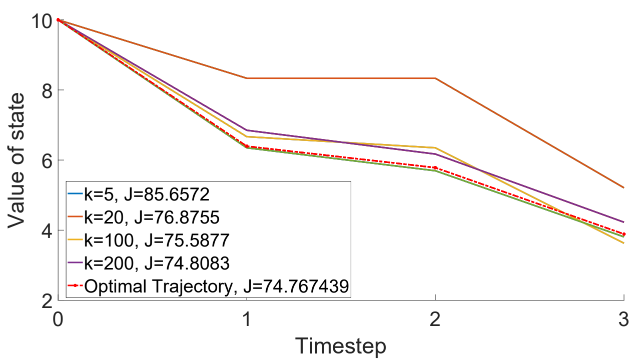

Figure 2. The resulting state trajectories from using the policy constructed from $P_k(t_0,x_0)$ in the Optimization Problem (28).

The Optimization Problem (28) can then be written in the form $A(t_0,x_0)$ using state augmentation,

$$\min z_3(4) \tag{29}$$

subject to,

$$z_1(t+1) = \frac{z_1(t)}{u(t)},\; z_2(t+1) = \begin{cases} u(t) \text{ if t=1} \\ 0 \text{ otherwise} \end{cases} \quad \forall t \in \{1,2,3\},$$

$$z_3(1) = u(1)^2, \quad z_3(2) = z_3(1) + u(1)^2,$$

$$z_3(3) = z_3(2) + z_2(2)^2 u(2), \quad z_3(4) = z_1(3)^2 \sqrt{z_3(3)} + z_3(3)^2,$$

$$z_1(0) = 10,\; z_2(0) = 0,\; z_3(0) = 0 \quad u(0), u(1), u(2) \geq 0.$$

The Optimization Problem (29) is now a special case of $P(t_0,x_0)$ and equivalent to the original Problem (28). The associated approximated optimization problem of the form $P_k(t_0,x_0)$ (12) can now be found by selecting appropriate compact state and control spaces; $X \subset \mathbb{R}^3$ and $U \subset \mathbb{R}$. A feasible policy for (28) is then constructed from the optimal policy of the associated $P_k(t_0,x_0)$ using (13). Figure 2 shows the state trajectories by following different constructed policies for various values of k. It is seen that for $k = 200$ the algorithm produces a solution within three significant figures of the analytic optimal objective function for (28).

IX. 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 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 $S(t_0,x_0)$ (5); which can be tractably solved as it has a NFSF as an objective function.

A. Battery Dynamics

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

$$e(k+1) = \alpha(e(k) + \eta u(k)\Delta t) \tag{30}$$

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 $q(k)$ as the power supplied by the grid at time step k.

$$q(k) = q_a(k) - q_s(k) + u(k) \tag{31}$$

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 $p_k = p_{on}$ if $k \in T_{on}$ and $p_k = p_{off}$ if $k \in T_{off}$ 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:

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

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:

$$J_D(\mathbf{u},\mathbf{e}) = p_d \max_{k\in\{t_{\text{on}},\ldots,t_{\text{off}}-1\}} \{q_a(k)-q_s(k)+u(k)\}.$$

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 $D(0,e_0)$.

$$\begin{aligned} &\min_{\mathbf{u},\mathbf{e}} \{J_E(\mathbf{u},\mathbf{e}) + J_D(\mathbf{u},\mathbf{e})\} \text{ subject to} \qquad (32) \\ &e(k+1) = \alpha(e(k)+\eta u(k)\Delta t) \text{ for } k = 0,\ldots,T \\ &e_0 = e0\ , e(k) \in X,\ u(k) \in U \text{ for } k = 0,\ldots,T, \\ &\mathbf{u} = (u(0),\ldots,u(T-1)) \text{ and } \mathbf{e} = (e(0),\ldots,e(T)) \end{aligned}$$

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

Proposition 28. *Problem $D(0,e_0)$ is a special case of $S(t_0,x_0)$*

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

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

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

D. Numerically Solving The Deterministic Battery Scheduling Problem

Our proposed approximation scheme can be applied to solve the battery scheduling problem, $D(0,e_0)$. This is done by creating an augmented state variable based on the maximum function in the objective function, as in Section III, and thus constructing an equivalent optimization problem of the form $A(0,x_0)$ (9); which is a special case of $P(t_0,x_0)$. Figure 3 shows how the monthly cost decreases when we use policies constructed from the associated discretized optimization problems, $P_k(t_0,x_0)$, and k is increased. Although we do not get a monotonically decreasing sequence of costs, the error

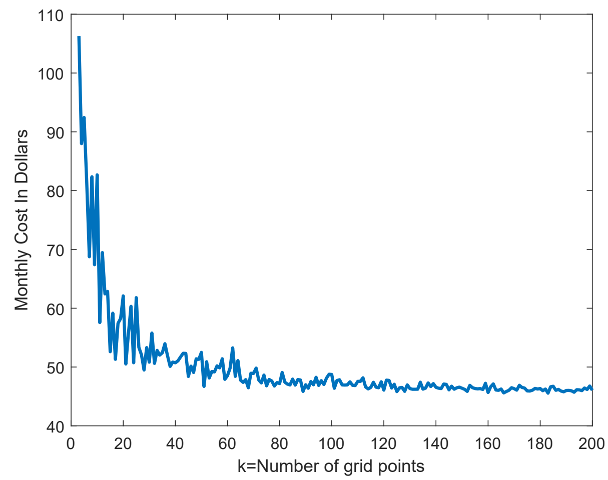

Figure 3. The resulting monthly cost from using the policy found by solving the discretized problem, of form $P_k(t_0,x_0)$, for optimal battery scheduling.

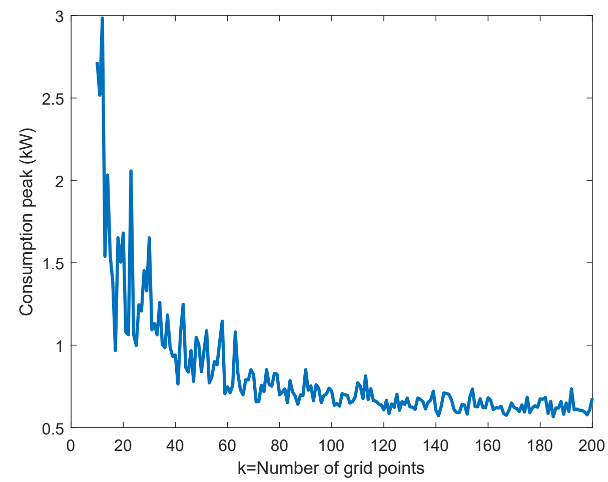

Figure 4. The resulting maximum demand from using the policy found by solving the discretized problem, of form $P_k(t_0,x_0)$, for optimal battery scheduling.

does decrease as $k \to \infty$. Figure 4 also shows that augmenting and then following our proposed discretization scheme for the battery scheduling problem results in a policy that reduces the consumption demand peak as k is increased. Figure 5 shows how the computational time required to solve the discretized battery scheduling problem appears to be of exponential nature with respect to the number of grid points.

We used solar and usage data obtained by local utility Salt River Project in Tempe, AZ, for power variables q_s and q_a. We also use pricing data from SRP for the parameters p_{on}, p_{off} and p_d. Battery data obtained for the Tesla Powerwall

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}	0.2973 ($/KWh)
$\bar{e}$	8000 (Wh)	Δt	0.5 (h)
t_{on}	27		

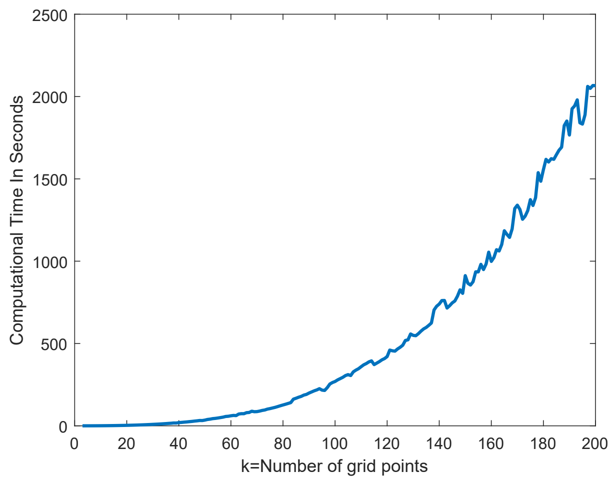

Figure 5. The computational time in seconds required to solve the discretized battery scheduling problem, of form $P_k(t_0,x_0)$.

was used to get the parameters α, η, $\bar{u}$, $\underline{u}$ and $\bar{e}$. The results of the simulation are shown in Figure 6. The policy used for this simulation was created using our augmentation and approximation scheme with $k = 20$. Interpolation was used to aid solving Bellam's equation (4) and decrease the approximation error. These results show an improvement in accuracy over results obtained based on the approach to a similar problem in [25] (approximately \$0.98 savings). As expected, we see the battery charges during off-peak and then discharges during on peak times to reduce ToU charges, while maintaining a reserve which it uses to keep consumption flat during on peak times, thereby minimizing the demand charge. As a result the power stabilizes during on peak times - becoming constant.

E. Solving the stochastic battery scheduling problem

To evaluate the effect of stochastic uncertainty on battery scheduling, we identified a Gauss-Markov model of solar generation based on SRP data. We construct the battery scheduling problem in the form $H_s(t_0,x_0)$ (15) and then use our proposed state augmentation approach to construct an equivalent optimization problem of form $Q(t_0,x_0)$ (16). The problem of form $Q(t_0,x_0)$ is then solved approximately using the methodology of Section VII-A.

F. Solar Generation Model

Our approach to modeling the dynamics of solar generation is based on [31]. Our Markov type model can be used to generate high resolution data over large time horizons. The Markov property of the model results in deviation from the mean being correlated time to time, helping represent the physical phenomena of clouds gradually passing over rather than instantaneously appearing.

Our model is a type of autoregressive-moving-average model (ARMAX) [32]. In [33] it is seen ARMAX models preform better than auto-regressive integrated moving average (ARIMA) and in [34] it is shown ARMAX models can produce data similar to real data for local sites in Califonia and Colorado.

Exogenous variables, temperature and humidity, are included as state variables in addition to the primary variable - solar radiance. Cross correlations between state variables are computed from data. 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:

$$\begin{aligned} &\mathbf{w}(t) = A\mathbf{w}(t-1) + B\mathbf{v}(t-1) \text{ for } t = 1,..,T \qquad (33)\\ &\text{where } \mathbf{w}(t) \in \mathbb{R}^3, \mathbf{w}(0) = \mathbf{0}\\ &\mathbf{v}(t) \sim \mathcal{N}(\mathbf{0}, I_{3\times3}), \end{aligned}$$

where the matrices A and 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 ([31])

$$A = M_1 M_0^{-1} \qquad BB^T = M_0 - M_1 M_0^{-1} M_1^T, \qquad (34)$$

where M_i is the i-lag cross correlation matrix. So $(M_i)_{m,n} = \rho_i(m,n)$ 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

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

Figure 7 shows simulated irradiance data from our solar model when compared to actual recorded irradiance data. For this numerical implementation the mean and standard deviation, $(\mu_i(t))_{0\le t\le T}$ and $(\sigma_i(t))_{0\le t\le T}$, were calculated using data from Wunderground for a weather station in Tempe, AZ on October the 15th 2014 for each state variable. Cross correlations between the variables were also calculated from the same data set and (34) was solved giving the matrices A and B in (33). As seen in the figure this solar generation model gives an output similar to what is observed in real data. In this particular instantiation the generated data lies mostly bellow the observed real data; however, this may not always be the case in further instantiations. Next we incorporate this model into our battery scheduling optimization problems.

Stochastic Battery Scheduling We now modify Problem $D(0,e_0)$ (32) to give a stochastic version of the battery scheduling problem $D_s(0,[e_0,0])$,

$$\begin{aligned} \arg\min_{\pi\in\Pi} \ \mathbb{E}_{[\mathbf{v}]_0^{T-1}} \Big[& J_E(\Phi_{f,0}(\pi,[e_0,0],T,[\mathbf{v}]_0^{T-1})) \qquad (35)\\ & + J_D(\Phi_{f,0}(\pi,[e_0,0],T,[\mathbf{v}]_0^{T-1})) \Big] \\ \text{subject to: } & \psi_{f,0}(\pi,[e_0,0],t,[\mathbf{v}]_0^{t-1}) \in E_t \times \mathbb{R}^3 \text{ for } t = 0,..,T\\ & \pi(x,t) \in U_t \text{ and } v(t) \in \mathbb{R}^3 \sim \mathcal{N}(\mathbf{0}, I_{3\times3}) \forall x \in X_t, \forall t = 0,..,T-1, \end{aligned}$$

where J_E is the ToU cost function and J_D is the demand charge found in Section IX-A; $f([e,w],u,t,v) = \begin{bmatrix} \alpha(e+\eta u\Delta t) \\ Aw+Bv \end{bmatrix}$; $E_t = [\underline{e},\bar{e}]$ and $U_t = [\underline{u},\bar{u}]$ for all $t \in \{0,...,T\}$; ψ_{f,t_0} and Φ_{f,t_0} are the state and trajectory map respectively defined in Definition 22; $[\mathbf{v}]_0^{T-1} = [v(0),..,v(T-1)] \in \mathbb{R}^{3\times(T)}$; matrices A and B are calculated from weather data using equations (34); and all constants are found in Table II.

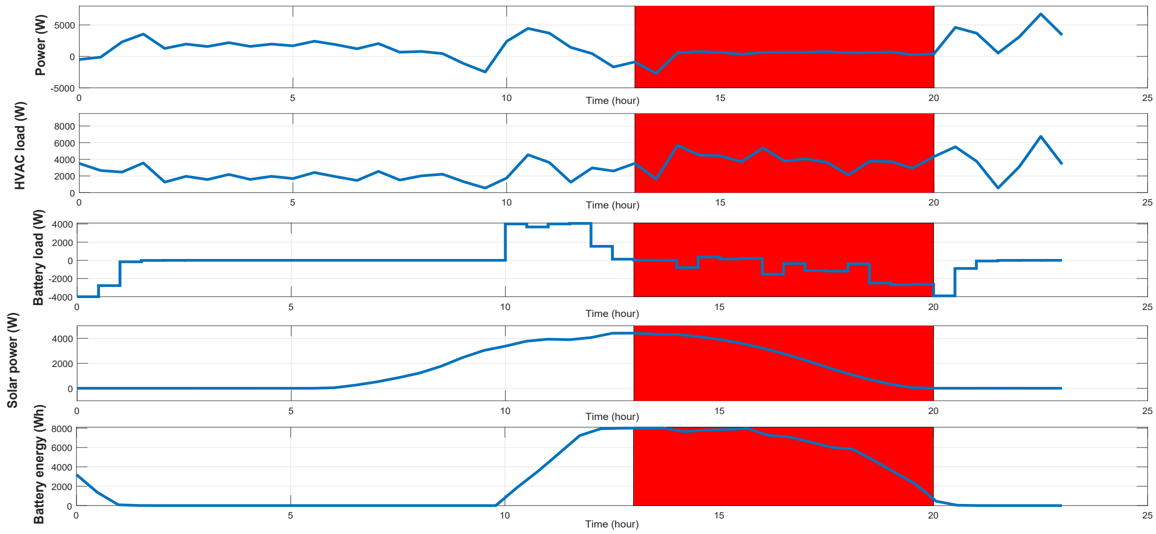

Figure 6. The trajectory the algorithm produces for deterministic solar data. The supremun of the power is 0.7033(kw) and the cost is $46.389.

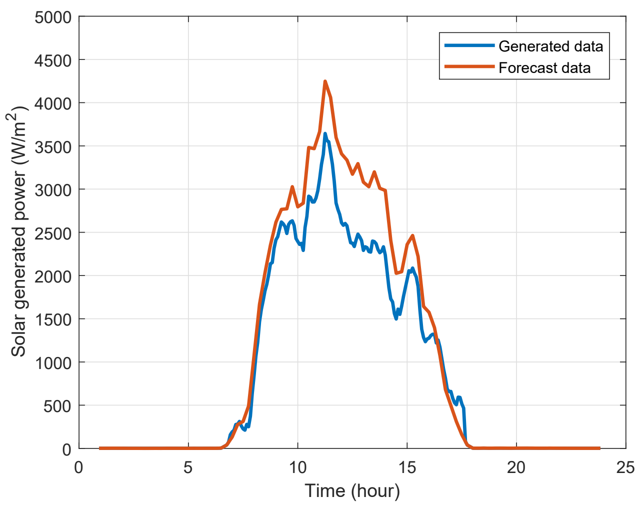

Figure 7. Solar data generated over 24 hours using data from Wunderground

G. Numerically Solving the Stochastic Battery Scheduling Problem

Using the state augmentation procedure in Section III on the stochastic battery scheduling problem $D_s(0,[e_0,0])$ (35), we may find an optimization problem of the form $Q(t_0,x_0)$ (16) such that the optimal policy for $D_s(0,[e_0,0])$ can be constructed form the optimal policy of $Q(t_0,x_0)$. We may then construct the approximated optimization problem $Q_{\varepsilon,k}(t_0,k)$ (25) and solve it using Bellman's equation (23). From the optimal policy of $Q_{\varepsilon,k}(t_0,k)$ we then construct a feasible policy for $Q(t_0,x_0)$ using (26). Figure 1 demonstrates a simulation of using the feasible policy obtained via augmenting and approximating the stochastic battery scheduling problem with a reasonably selected family of compact state spaces, $\{H_{\varepsilon,t}\}_{0\le t\le T}$, and discretization level $k=10$. To simply computation we used a one state version of our solar model (33) and used interpolation while solving Bellman's equation. As expected the battery charges during the on peak times and conservatively discharges during the off-peak times. The solar data generated from this run were then used as input to the deterministic algorithm in order to compare performance. As anticipated, the deterministic case performs better than the stochastic case.

X. Conclusion

In this paper we have proposed a generalized formulation of the DP problem and shown that if the objective function is forward separable, such problems may reformulated using state augmentation with an equivalent DP problem with additively separable objective function. Furthermore, we have defined a class of functions, called naturally forward separable functions, such that DP problems with an objective function of this class can be tractably solved using state augmentation. Moreover, 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 DP problems. We have further extended these results to stochastic DP 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.

XI. Appendix

Lemma 29. *Consider an optimization problem of the form $Q(t_0,x_0)$ (16) with additively separable objective function $J^Q_{t_0,x_0}$. For any family of functions of the form $\hat{\pi}: \mathbb{R}^{n\times(t-t_0+1)} \to \mathbb{R}^m$ are such $\hat{\pi}_t([(x(t_0),\dots,x(t))]) \in U_t$ and $f(x(t),\hat{\pi}_t([(x(t_0),\dots,x(t))]),t,v(t)) \in X_{t+1}$ for all $x(i)\in X_i$, $i\in\{t_0,\dots,t\}$, $v(t)\in\mathbb{R}^q$ and $t\in\{t_0,\dots,T-1\}$ there exists $\alpha\in\Pi$ such that*

$$\mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\alpha,x_0,T,[\mathbf{v}]_{t_0}^{T-1}))\right) = \mathbb{E}_{[\mathbf{v}]_{t_0}^{T-1}}\left(J^Q_{t_0,x_0}(\Phi_{f,t_0}(\hat{\pi},x_0,T,[\mathbf{v}]_{t_0}^{T-1}))\right) \quad (36)$$

where we make a small abuse of notation to extend the trajectory map Φ_{f,t_0} to policies that use the entire state space history.

Proof. Proposition 8.1 [35] or Theorem 6.2 [4]. □

Acknowledgments

This research was supported by the NSF Grant CNS-1739990. We gratefully acknowledge data and expertise provided by Robert Hess of Salt River Project.

References

[1] R. E. Bellman, R. E. Kalaba, and T. Teichmann, "Dynamic programing," *Physics Today*, vol. 19, p. 98, 1966.

[2] R. Weber, *Optimization and Control*. 2016.

[3] M. Sniedovich, *Eureka! Bellman's principle of optimality is valid!* Kluwer.

[4] K. Hinderer, *Foundations of Non-stationary Dynamic Programming with Discrete Time Parameter*. Springer-Verlag.

[5] R. H. Strotz, "Myopia and inconsistency in dynamic utility maximization," *The Review of Economic Studies*, vol. 23, no. 3, pp. 165–180, 1955.

[6] A. Shapiro, "On a time consistency concept in risk averse multistage stochastic programming," *Operations Research Letters*, vol. 37, no. 3, pp. 143–147, 2009.

[7] P. Carpentier, J.-P. Chancelier, G. Cohen, M. De Lara, and P. Girardeau, "Dynamic consistency for stochastic optimal control problems," *Annals of Operations Research*, vol. 200, no. 1, pp. 247–263, 2012.

[8] 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.

[9] A. Domingo and M. Sniedovich, "Experiments with dynamic programming algorithms for nonseparable problems," *European Journal of Operational Research*, vol. 67, pp. 172–187, 1993.

[10] E. Lourens, E. Reynders, G. De Roeck, G. Degrande, and G. Lombaert, "An augmented kalman filter for force identification in structural dynamics," *Mechanical Systems and Signal Processing*, vol. 27, pp. 446–460, 2012.

[11] J. Skaf, S. Boyd, and A. Zeevi, "Shrinking-horizon dynamic programming," *International Journal of Robust and Nonlinear Control*, vol. 20, no. 17, pp. 1993–2002, 2010.

[12] N. Bäuerle and U. Rieder, "More risk-sensitive markov decision processes," *Mathematics of Operations Research*, vol. 39, no. 1, pp. 105–120, 2013.

[13] W. B. Haskell and R. Jain, "A convex analytic approach to risk-aware markov decision processes," *SIAM Journal on Control and Optimization*, vol. 53, no. 3, pp. 1569–1598, 2015.

[14] J. Rust, "Using randomization to break the curse of dimensionality," *Econometrica: Journal of the Econometric Society*, pp. 487–516, 1997.

[15] F. Dufour and T. Prieto-Rumeau, "Approximation of Markov decision processes with general state space," *J. Math. Anal. Appl.*, vol. 388, pp. 1254–1267, 2012.

[16] M. Jones and M. M.Peet, "A dynamic programming approach to evaluating multivariate gaussian probabilities," 2018.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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*, pp. 5348–5353, IEEE, 2016.

[26] R. Kamyar and M. M. Peet, "Optimal thermostat programming and optimal electricity rates for customers with demand charges," in *American Control Conference (ACC), 2015*, pp. 4529–4535, IEEE, 2015.

[27] M. Jones and M. M. Peet, "Solving dynamic programming with supremum terms in the objective and application to optimal battery scheduling for electricity consumers subject to demand charges," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*, pp. 1323–1329, IEEE, 2017.

[28] O. Hernandez-Lerma and J. Lasserre, *Discrete-Time Markov Control Processes: Basic Optimality Criteria*, vol. 42. 1999.

[29] D. Li and Y. Y. Haimes, "The envelope approach for multiobjeetive optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 17, no. 6, pp. 1026–1038, 1987.

[30] E. Porteus, "An informal look at the principle of optimality," *Managment Science*, vol. 21, 1975.

[31] C. W. Richardson, "Stochastic simulation of daily precipitation, temperature, and solar radiation," *Water resources research*, vol. 17, no. 1, pp. 182–190, 1981.

[32] C. Wan and Z. Hu, "Photovoltaic and solar power forecasting for smart grid energy management," *Journal of power and energy systems*, vol. 1, 2015.

[33] Y. Li and L. Shu, "An ARMAX model for forecasting the power output of a grid connected photovoltaic system," *Renewable Energy*, vol. 60, pp. 78–89, 2014.

[34] Y. Chen and L. Xie, "Multitime-scale data-driven spatiotemporal forecast of photovoltaic generation," *IEEE Transactions on Sustainable Energy*, vol. 6, pp. 104–112, 2015.

[35] D. Bertsekas and S. Shreve, *Stochastic Optiamal Control: The Discrete-Time case*. Athena Scientific, 1978.

Morgan Jones received the B.S. and Mmath in mathematics from The University of Oxford, England in 2016. He is a research associate with Cybernetic Systems and Controls Laboratory (CSCL) in the School for Engineering of Matter, Transport and Energy (SEMTE) at Arizona State University. His research primarily focuses on the estimation of attractors, reachable sets and regions of attraction for nonlinear ODE's. Furthermore, He has been studying the effects of optimal energy storage in smart grid environments.

Matthew M. Peet received the B.S. degree in physics and in aerospace engineering from the University of Texas, Austin, TX, USA, in 1999 and the M.S. and Ph.D. degrees in aeronautics and astronautics from Stanford University, Stanford, CA, in 2001 and 2006, respectively. He was a Postdoctoral Fellow at INRIA, Paris, France from 2006 to 2008. He was an Assistant Professor of Aerospace Engineering at the Illinois Institute of Technology, Chicago, IL, USA, from 2008 to 2012. Currently, he is an Associate Professor of Aerospace Engineering at Arizona State University, Tempe, AZ, USA. Dr. Peet received a National Science Foundation CAREER award in 2011.