Addressing the stochastic nature of energy management in smart homes

Chanaka Keerthisinghe, Student MIEEE, Gregor Verbič, Senior MIEEE, and Archie C. Chapman, MIEEE
School of Electrical and Information Engineering
The University of Sydney
Sydney, New South Wales, Australia
Email: {chanaka.keerthisinghe, gregor.verbic, archie.chapman}@sydney.edu.au

Abstract—In the future, automated smart home energy management systems (SHEMSs) will assist residential energy users to schedule and coordinate their energy use. In order to undertake efficient and robust scheduling of distributed energy resources, such a SHEMS needs to consider the stochastic nature of the household’s energy use and the intermittent nature of its distributed generation. Currently, stochastic mixed-integer linear programming (MILP), particle swarm optimization and dynamic programming approaches have been proposed for incorporating these stochastic variables. However, these approaches result in a SHEMS with very costly computational requirements or lower quality solutions. Given this context, this paper discusses the drawbacks associated with these existing methods by comparing a SHEMS using stochastic MILP with heuristic scenario reduction techniques to one using a dynamic programming approach. Then, drawing on analysis of the two methods above, this paper discusses ways of reducing the computational burden of the stochastic optimization framework by using approximate dynamic programming to implement a SHEMS.

Keywords—future grid, demand response, smart home, stochastic mixed-integer linear programming, dynamic programming, scenario reduction techniques, approximate dynamic programming.

I. INTRODUCTION

FUTURE electrical power grids will be based on digitalized information and communication technologies that enable bi-directional flow of information and electrical power. Also referred to as the “smart grid”, the aim is more efficient, reliable and green electrical power system. The development of this future grid is expected to enable utilities and customers to seize the full potential of demand side management (DSM) and demand response (DR) programs [1]. On the one hand, utilities are looking at DSM and DR programs to better manage their electricity networks; while on the other hand, DR programs encourage customers to reduce loads during periods of critical network congestion or periods of high energy prices, in return reducing their electricity costs. Given this context, residential and commercial buildings will play a major role in any DR program, and this paper focuses on residential buildings.

In particular, a smart home is envisioned as an automated residential building that uses distributed energy resources (DERs) for managing energy consumption and providing suitable levels of comfort to the end-user. As depicted in Fig. 1, DERs in a smart home can consist of: distributed generation (DG) units, such as rooftop photovoltaic systems (PVs), fuel cells and micro-turbines; energy storage units, such as a battery storage, electric vehicle (EV) battery storage and thermal energy storage (TES) units (i.e. hot water tank); and end-user loads, such as cyclical loads (i.e. heating, ventilation and air conditioning (HVAC) system, refrigerator and hot water heater) and shiftable loads (i.e. washing machine, dryer and dishwasher).

A smart home energy management system (SHEMS) can help a customer seize the full financial benefits of the available DERs. However, in order to undertake efficient and robust scheduling of DERs, a SHEMS needs to consider the stochastic nature of the household’s energy use and the intermittent nature of its DG, which includes electrical power generation from DG units, electrical power demand, thermal energy demand, availability of an electric vehicle (EV) as storage and the required end-of-day battery state of charge (SOC). Incorporating these stochastic variables using appropriate probability distributions is an important aspect of an effective SHEMS.

Currently, proposed methods for implementing a SHEMS include: mixed-integer linear programming (MILP) [2]–[4], particle swarm optimization (PSO) [5]–[7] and dynamic programming approaches [8]. As presented in [2]–[4], MILP is widely used in SHEMSs mainly because of the simplicity associated with off-the-shelf MILP solvers, such as MOSEK and CPLEX. In [5]–[7], PSO is used for robust scheduling of the DERs in a smart home. However, neither MILP nor PSO considers the uncertainties in a household’s energy use and the intermittent nature of its DG.

A comparison of a stochastic and deterministic optimizations using MILP is presented in [9]. The paper concluded that stochastic optimization has the better quality solutions as it considers the uncertainties in the household. However, the energy management problem becomes computationally challenging. Robust scheduling of DERs with stochastic programming using PSO is presented in [7], which is only preliminary work and the accuracy of the results depends on the selection method used for reducing the number of scenarios. A dynamic programming approach for a SHEMS presented in [8] considers the stochastic variables of the household. However, the computational complexity increases exponentially with dimensionality of state, outcome and action spaces, making it impractical in large-scale or long-horizon problems.

Within this context, this paper discusses the drawbacks associated with these existing methods, such as the computational burden and the quality of the solutions. In order to do this, a SHEMS that uses stochastic MILP with heuristic scenario reduction technique is compared to a one that uses a
dynamic programming approach.

The paper is structured as follows: Section II states the general optimization problem and describes the stochastic variables of the household. This is followed by a description of the stochastic MILP and dynamic programming formulations in Section III and IV, respectively. Section V consists of a description of the two implemented SHEMSs. Then, the simulation results are used to illustrate the drawbacks associated with the existing methods in Section VI. Drawing on the analysis of the simulation results, Section VII presents ways of reducing the computational burden of the stochastic optimization framework by using approximate dynamic programming methods for implementing a SHEMS.

II. GENERAL FORMULATION OF THE ENERGY MANAGEMENT PROBLEM

The smart home considered in this paper is depicted in Fig. 1, and consists of a PV system, micro-combined heat and power (MCHP) fuel cell system, battery storage, EV and TES unit. The decision horizon is a 24 hour period starting at 6 : 30 am, divided into 15 minutes time-steps, giving a total of $N = 96$ time-steps, where $k$ denotes a particular time-step.

A. Optimization problem

Depending on the customer’s choice, the objective function of the optimization problem can consider different criteria, such as energy costs, user comfort preferences, energy consumption, CO2 emissions costs and peak load. The general formulation of the objective function used in this paper is represented using control/decision and state variables.

Let:

$$x^i = \begin{bmatrix} x^i_1 & \cdots & x^i_{|S|} \end{bmatrix}^T, \quad \text{and} \quad s^i = \begin{bmatrix} s^i_1 & \cdots & s^i_{|S|} \end{bmatrix}^T,$$

where $x^i$ and $s^i$ are the decision and state variables, respectively, of device $i \in S$ over the decision horizon, for all $k \in \{1 \ldots N\}$. Furthermore, let $X$ and $S$ be matrices of decision and state variables, respectively, for all devices; that is:

$$X = \begin{bmatrix} x^1 & \cdots & x^{|S|} \\ \vdots & \ddots & \vdots \\ x^N & \cdots & x^{|S|} \end{bmatrix}, \quad \text{and} \quad S = \begin{bmatrix} s^1 & \cdots & s^{|S|} \\ \vdots & \ddots & \vdots \\ s^N & \cdots & s^{|S|} \end{bmatrix}.$$

Formally, the problem is given by:

$$\begin{align*}
\text{minimize} & \quad f_o(X, S), \\
\text{subject to} & \quad x^i_k \ \text{and} \ s^i_k \ \text{satisfy device and user constraints,} \\
& \quad \forall k \in \{1 \ldots N\}, \\
& \quad \forall i \in S.
\end{align*}$$

(1)

The state variable, $s^i_k \in S$, contains the minimum necessary and sufficient information used to compute the decision/control variable, $x^i_k \in X$, which is a control action for each device, $i \in S$, over the decision horizon. Device $i$’s constraints consist of power limits and mathematical models describing the operational characteristics. The user comfort criteria consider variables, such as the internal temperature of the household, and whether the shiftable loads are completed on time.

B. Stochastic variables

Currently, proposed SHEMSs require accurate weather forecasts and an exact knowledge about the behavioral patterns of the inhabitants to generate efficient schedules for DERs. However, making such forecasts and predictions is difficult to do in the real world. Moreover, as explained in [7], inaccurate predictions may result in additional costs, and may affect the comfort of the inhabitants. Therefore, incorporating these uncertainties using appropriate probabilistic models is an important aspect of an effective SHEMS.

1) PV output: PV output depends on weather patterns, and a forecast is assumed to be available before the horizon starts. Even though the forecast may be fairly accurate, uncertainties in transitions of the weather (i.e. going cold or sunny from the given prediction) need to be considered.

2) Electrical demand of the household: Electrical demand depends on the number of occupants and their behavioral patterns. However, modeling these behavioral patterns is difficult [10]. Therefore, it is often assumed that the electrical demand is directly related to the number of occupants in the household at a given time. As such, the probabilities for occupancy transitions (i.e. all home, some home and all away) should also be considered.

3) Thermal demand of the household: Thermal demand is typically considered to be primarily related to the external temperature. Therefore, transition probabilities for ambient temperature should be considered.

4) Availability of the EV: It is difficult to mathematically model the tendency of the inhabitants to use their EVs. Typically, it is assumed that the occupants have a fair knowledge about the transition probabilities associated with the availability of the EV as storage.

5) The required end of the day battery SOC: As the PV output is one of the main sources of energy in a smart home, the required battery SOC at the end of the day should depend on the PV output and the electrical demand in future days. As such, controlling battery SOC will be a special focus of attention in Section VII-B of this paper.
III. STOCHASTIC MILP APPROACH

Deterministic versions of the energy management problem are often solved using a MILP approach, which is an optimization of a linear objective function subject to linear constraints with continuous and integer variables [9]. However, this method can also be applied to the stochastic optimization problem formulated in Section II by using a scenario-based approach, which is referred to as stochastic MILP.

In order to use a stochastic MILP approach, the device i’s operational constraints are linearized and a large number of scenarios is generated by sampling from possible combinations of the stochastic variables mentioned in Section II-B. A larger number of scenarios should improve the solutions generated by better incorporating the stochastic variables, but this imposes a greater computational burden. Therefore, heuristic scenario reduction techniques are employed to obtain a scenario set of size J, which can be solved within a given time with reasonable accuracy.

A scenario-based stochastic MILP formulation of the problem is described by:

\[
\minimize \sum_{j=1}^{J} P_j(s^i) f_o(X, S),
\]

where \( P_j(s^i) \) is the probability of a particular scenario \( j \) corresponding to realizations of stochastic variables \( s^i \). The sum of the probabilities of all the scenarios in \( J \) adds up to 1:

\[
\sum_{j=1}^{J} P_j(s^i) = 1.
\]

IV. DYNAMIC PROGRAMMING

The stochastic energy management problem formulated in Section II can be solved using dynamic programming, since the problem is a discrete time dynamic system (i.e. can be modeled as Markov decision processes) with a cost function that is additive over time. In this section, Markov decision processes are described, value functions and optimality conditions are explained and a method to obtain the optimal policy is presented.

A. Markov decision processes (MDPs)

A MDP is a discrete time state-transition system described by: a state space, \( (s \in S) \); a decision space, \( (x \in X) \); transition functions, \( P(s'|s, x) \), describing the probability of landing on state \( s' \), given action \( x \) from state \( s \) after a realization of random variables, \( \omega_k \); and a contribution (i.e. reward or cost) function, \( C_k(s, x) \), which is an instantaneous utility function that describes how good/bad it is to take an action in a particular state. It is often better to collect the rewards sooner than later, so a discount factor, \( \gamma \), is used to account for this in some problems; \( \gamma \in (0, 1] \).

B. Value functions and optimality conditions

Dynamic programming is used to solve the optimization problem of the form:

\[
\min_{\pi} \mathbb{E} \left\{ \sum_{k=0}^{N-1} \gamma^k C_k^o(s_k, x_k, \omega_k) + \gamma^N C_N^o(s_N) \right\},
\]

where: \( \pi \) is a policy, a choice of action for each state, \( \pi : S \rightarrow X \); \( C_N^o(s_N) \) is a terminal cost incurred at the end of the horizon; and \( C_k^o(s_k, x_k, \omega_k) \) is the cost incurred at a given time-step \( k \), which accumulates over time. Therefore, the cost function is additive, and the problem is formulated as an optimization of the expected cost because the cost is generally a random variable due to the effect of \( \omega_k \) [11].

The expected future discounted cost of following a policy, \( \pi \), starting in state, \( s_k \), is \( V^\pi(s_k) \), which is given by the value function:

\[
V^\pi(s_k) = \sum_{s' \in S} \mathbb{P}(s_k, \pi(s_k), s') \left[ C(s_k, \pi(s_k), s') + \gamma V^\pi(s') \right].
\]

(5)

C. Optimal policy

An optimal policy, \( \pi^* \), is one that minimizes (5). It can be found recursively by computing the optimal value function, \( V_k^*(s_k) \), using Bellman’s optimality condition:

\[
V_k^*(s_k) = \min_{x_k \in X_k} \left( C_k^o(s_k, x_k) + \gamma \mathbb{E} \left\{ V_{k+1}^*(s_{k+1}) | s_k \right\} \right).
\]

(6)

The expression in (6) is easily computed using backward induction, a procedure called value iteration, and then the optimum policy can be extracted from the value function by selecting the minimum value action for each state.

V. IMPLEMENTED SMART HOME ENERGY MANAGEMENT SYSTEMS

In this section, the stochastic energy management problem described in Section II is solved using the two solution methods, stochastic MILP and dynamic programming, which are explained in Section III and IV, respectively.

A. Formulation of the problem

The smart home consists of a PV system, fuel cell MCHP system, battery storage, EV and TES unit. Specifically:

1) State variables are used to represent the battery SOC, \( s_k^{B} \), TES unit state, \( s_k^{T} \), fuel cell output, \( s_k^{FC} \), availability of the EV, \( s_k^{EV,a} \) (i.e. a binary variable), and EV battery SOC, \( s_k^{EV} \), for each time-step, \( k \), in the decision horizon.

2) Control variables consists of charge rate of the battery, \( x_k^{B} \), discharge rate of the battery, \( x_k^{B^-} \), electrical energy generated by fuel cell, \( x_k^{FC,c} \), thermal energy generated by fuel cell, \( x_k^{FC,t} \), charge rate of the EV battery, \( x_k^{EV} \), discharge rate of the EV battery, \( x_k^{EV^-} \), thermal energy generated by the supplementary water heater, \( x_k^{WH} \), distribution of DC energy between battery and inverter, \( x_k^{DC} \), and electrical grid power, \( x_k^{grid} \), at a given time-step \( k \).

3) Transition functions govern how the state variables evolve over time. For example, the SOC of the battery described by the state variable \( s_k^{B} \in \{ s_k^{B,min}, s_k^{B,max} \} \) will progress by:

\[
s_k^{B+} = (1 - l_k^{B}(s_k^{B}))(s_k^{B} - x_k^{B^-} + \mu_k^{B+}(x_k^{B+})),
\]

(7)

where \( l_k^{B}(s_k^{B}) \) is the efficiency of the charging process and \( l_k^{B}(s_k^{B}) \) models the self-discharging process of the battery.
4) Random variables are PV output, \( \omega_k^{PV} \), electrical demand, \( \omega_k^{D,e} \), and thermal demand, \( \omega_k^{D,t} \).

5) All parameters used are given in Table I, and transition probabilities, forecasts and characteristics of the devices are given in Fig. 2. The input characteristics of the devices indicate that dynamic programming can properly incorporate non-linear characteristics, while linear approximations have to be made with stochastic MILP.

**B. Objective function**

The main objective of the implemented SHEMS is to minimize energy costs over the decision horizon, \( \forall k \{1 \ldots N \} \), which consists of cost of electricity consumption, \( p_k^{g,s} \), revenue from selling electrical power back to the grid, \( p_k^{g,s} \) and cost of gas consumption, \( p_k^{gas} \). The SHEMS is also capable of transferring the excess power back to the electrical grid during peak periods, which can generate an income. This is given by transferring the excess power back to the electrical grid during peak periods, which can generate an income. This is given by:

\[
\text{Electricity sold to the electrical grid is much lower than the peak periods, which can generate an income.}
\]

As mentioned in Section III, in stochastic MILP, heuristic scenario reduction techniques are employed to obtain the desired scenario set, \( J \). Specifically, the techniques used are backward and forward scenario reduction, which are already used to obtain optimal scenario trees in power systems; a detailed explanation of these can be found in [7].

**VI. Simulation Results**

This section presents analysis of the simulation results obtained for the two implemented SHEMSs. The SHEMSs implemented using the stochastic MILP and the DP approaches are operating according to the following constraints:

- Fuel cell is only turned on with the highest possible output of electrical power.
- Charge the battery with the highest possible charge rate if the combined power generated from the fuel cell and the PV system exceeds the household demand.
- Only transfer power back to the electrical grid if the combined power generated by the household exceeds the household demand and the highest possible charge rate of the batteries. This is because feed-in tariffs (FiTs) are usually less than the retail tariffs paid by the households. For example: in Australia it is \( \approx 8c/kWh \) [12].

The total daily financial costs caused by the SHEMS using dynamic programming is \( \$2.48 \) while \( \$2.72 \) for the SHEMS using stochastic MILP. In Fig. 3, a comparison is made of the key state variables. These indicate that the dynamic programming approach results in a better quality as it uses scenario reduction technique to reduce the size of the scenario set, so that the problem can be

**C. Computational aspects**

In both of the implemented SHEMSs, the values of the control variables \( x_k = [x_k^{FC,e}, x_k^{A,DC}, x_k^{H}, x_k^{EV^+}], \) are to be determined depending on the state variables \( s_k = [s_k^{FC}, s_k^{A}, s_k^{B}, s_k^{EV}, s_k^{EV^+}, s_k^{T}, s_k^{EV,t}, s_k^{EV,s}]. \) and realizations of random variables, at each time-step.
solved within a desired time limit with reasonable accuracy. The financial costs caused by full and half scenarios are $2.72 and $2.91, respectively, validating the value of the scenario reduction technique. Furthermore, a comparison of the solutions are presented for a full and half number of scenarios in Fig. 3.

VII. APPROXIMATE DYNAMIC PROGRAMMING AND FUTURE RESEARCH DIRECTIONS

This section draws on the analysis presented above to identify directions for future work. This will involve using approximate dynamic programming (ADP) methods to solve the optimization problem formulated in Section II and IV, with the aim of improving the computational performance over that of the dynamic programming, but with a similar
solution quality. The conjecture is that approximate dynamic programming can:

- scale to handle fine-grained temporal and spatial mod-

eling, which avoids looping over all possible states, actions and outcomes,
- model the shiftable/controllable user loads in a smart home with less computational effort,
- use a two-stage optimization, as mentioned in Section VII-B, but with a higher resolution (i.e. 15 minutes time-steps) that is computationally feasible,
- easily incorporate on-line learning methods mentioned in VII-C, which will enable seamless integration of DERs in a plug-and-play manner, increasing ancillary services to the electrical grid.

In the proposed approximate dynamic programming approach, the optimization problem in (4) is solved by computing the optimal value function (6) while stepping forward in time. Since the values of the future states are not yet computed, policies (i.e set of actions) are computed by value function approximation and/or policy function approximations, as explained in Section VII-A and C, respectively.

A. Policy based value function approximation

One approach in ADP is to approximate the value function and extract a policy from this approximation. This can be found iteratively, as mentioned in [13]. However, approximating the expectation within the max or min operator is difficult to do in practical applications. Here the focus is on approximating the value function around a post decision state vector, $s_k^T$, the state of the system at discrete time, $k$, soon after making the decisions but before the realization of random variables. Hence the new form of the value function is written as:

$$V_k^\pi(s_k) = \min_{x_k} \left( C_k(s_k, \pi(s_k)) + \gamma^k V_{k+1}^{\pi,x}(s_{k+1}^T) \right), \tag{11}$$

where $V_{k+1}^{\pi,x}(s_{k+1}^T)$ is the value function approximation around the post-decision state $s_{k+1}^T$, given by:

$$V_{k+1}^{\pi,x}(s_{k+1}^T) = \mathbb{E} \left\{ V_{k+1}(s_{k+1}) \mid s_k^T \right\}. \tag{12}$$

This method is computationally feasible because $\mathbb{E} \left\{ V_{k+1}^{\pi}(s_{k+1}^T) \right\}$ is a function of the post-decision state $s_{k+1}^T$ (i.e a deterministic function of $x_k$) rather than $s_k$ in the original value function in (6). In order to approximate the value function around a post decision state vector: (i) start with the initial value function approximations and step forward in time by solving the problem at each time-step, using the value function approximation of the future state; and, (ii) repeat this procedure over several iterations, updating the value function approximation after each iteration. Each iteration depends on a sample realization of the random information, a process that is referred to as Monte Carlo simulation. This iterative updating enables to obtain an approximation of the value function around the post decision state [14].

The initial value function approximations for post decision states can be found by solving a simplified optimization problem using off-the-shelf solvers, as explained in Section VII-B and/or policy function approximations, as explained in Section VII-C.
B. Approximating the values of state variables

In finite horizon problems, accurately estimating the value of the state variable in the final time-step can improve the solution quality. In this section, this concept is illustrated with a two-stage look-ahead optimization of the energy management problem. As explained in Section II, the required battery SOC at the end of the horizon, $s_{N}^{B}$, should depend on the PV output and the predicted demand in future days. Given this context, a simplified optimization over a longer horizon (i.e weekly) is carried out considering the weekly PV and demand forecast to approximate the value of $s_{N}^{B}$.

As shown in Fig.4, the battery SOC is maximal when the demand is high, and end-of-day battery SOC depends on the PV output and the electrical demand in future days (i.e. end-of-day one battery SOC is minimized since day two and day three are sunny with low demand). The approximated value of the end-of-day battery SOC, $s_{N}^{B}$, can be used in a more detail stochastic daily optimization, which improves the quality of the solutions. For example: if the required end-of-day battery SOC is 5 kWh instead of 9.5 kWh, as depicted in Fig. 3, the financial costs are reduced by $0.28, which can be possible if future days are sunny with low demand. Finally the operating schedules (i.e. set of decisions) from a deterministic daily optimisation can be used to approximate the initial value functions and post decision states (mentioned in Section VII-A).

C. Policy function approximations

For the sake of completeness, policy function approximations are also described, which are functions that return an action for a given state. However, designing specific functions that map a state to an action is a challenge, which can be overcome by three strategies: (i) rule-based lookup table with actions for each state or a value for each state; (ii) parametric models used to obtain value of being in each state depending on different conditions. For example: the battery can be charged if electricity costs are below a certain value and discharge if it is above a certain value; and (iii) nonparametric statistical representations, such as Kernel regression, support vector regression and neural networks (i.e. learning methods). An interested reader can refer to [14] for more detail.

VIII. Conclusion

This paper presents a comparison of two existing solution methods, stochastic MILP and dynamic programming, used to schedule DERs in a smart home. The drawbacks associated with these existing SHEMSs were identified in order to propose future research work using approximate dynamic programming methods, which is expected to improve the computational performance of dynamic programming but with similar solution quality. In particular, a two-stage look-ahead optimization of a smart home energy management problem was used as an example to show how approximate dynamic programming methods can be applied, which will be considered in future research.