Joint Control of Manufacturing and Onsite Microgrid System via Novel Neural-Network Integrated Reinforcement Learning Algorithms

Wenqing Hu, Zeyi Sun, Jiaojiao Yang, Louis Steimeister, and Kaibo Xu

Abstract—Microgrid is a promising technology of distributed energy supply system, which typically consists of storage devices, generation capacities including renewable sources, and controllable loads. It has been widely investigated and applied for residential & commercial end use customers as well as critical facilities. In this paper, we propose a joint dynamic control model of microgrids and manufacturing systems using Markov Decision Process (MDP) to identify an optimal control strategy for both microgrid components and manufacturing system so that the energy cost for production can be minimized without sacrificing production throughput. The proposed MDP model has a high dimensional state/action space and is complicated in that the state and action spaces have both discrete and continuous parts and are intertwined through constraints. To resolve these challenges, a novel reinforcement learning algorithm that leverages both on-policy temporal difference control (TD-control) and deterministic policy gradient (DPG) algorithms is proposed. In this algorithm, the values of discrete decision actions are first learned through neural network integrated temporal difference iteration, while the parameterized values of continuous actions are learned from deterministic policy gradients. The constraints are then addressed via proximal projection operators at the policy gradient updates. Experiments for a manufacturing system with an onsite microgrid with renewable sources have been implemented to identify optimal control actions for both manufacturing system and microgrid components towards cost optimality. The experimental results show the effectiveness of combining TD control and policy gradient methodologies in addressing the “curse of dimensionality” in dynamic decision-making with high dimensional and complicated state and action spaces.

Index Terms—Microgrid, Manufacturing System, Reinforcement Learning, Markov Decision Process, On-Policy Temporal Difference Learning, Deterministic Policy Gradient

I. INTRODUCTION

A microgrid is a localized autonomous energy system that consists of distributed energy sources and loads, which can operate either separated from, or connected to, external utility power grids [1], [2], [3]. It is considered a reliable solution to satisfy the growing demand of electric power through strengthening the resilience and mitigating the disturbances of the grid [4], [5], [6].

Various studies on microgrids have been conducted for residential houses [7], [8], [9] and critical facilities, such as medical centers, financial corporations, military bases, and jails [10], [11]. While, since manufacturing is traditionally not considered a critical facility, the research on the application of microgrid in manufacturing has been less reported.

However, manufacturing activities dominate energy consumption and green-house-gas (GHG) emissions in the industrial sector [12] that accounts for one third of the total energy consumption in the U.S. [13]. Furthermore, it is hardly possible to maintain manufacturing operations without electricity supply nowadays, even a very short power outage can lead to detrimental impacts on manufacturing companies [14], [15], [16], [17].

Thus, the research focusing on the optimal design and component sizing of the microgrid for manufacturing plant has been recently launched [18], [19], [20]. For example, a Mixed Integer Non-Linear Programming optimization model was proposed for sizing the capacity of onsite generation system with renewable sources and battery energy storage system for the manufacturers considering the energy loads from both manufacturing system and HVAC system in a typical manufacturing plant [18].

In addition, optimal energy control from the manufacturing side has also been widely investigated [21], [22], [23], [24]. For example, a simulation based model was proposed to investigate the energy control of manufacturing system in demand response program [21]. An analytical model was later proposed to identify an optimal production schedule for typical manufacturing systems in Time-of-Use demand response program [22].

However, the study of joint energy control and management for both on-site microgrid generation system and manufacturing plant simultaneously has not yet been fully launched. The major challenges impeding the research in this area can be summarized from two aspects, i.e., modeling and solution. On one hand, the combined system including both manufacturing and microgrid has a complex interaction when controls are implemented from both sides. For example, the controls on manufacturing systems will influence the energy demand that needs to be met through controlling the operations of microgrids to achieve an energy flow balance. The
manufacturing system itself is a complex system where the interrelationships among different machines in the system need to be quantified. The manufacturing throughput should not be sacrificed when energy control for the manufacturing system is implemented. All these factors need to be carefully considered when modeling the decision-making using Markov Decision Process (MDP).

On the other hand, it can be expected that the space of the states and the actions in the MDP model will be very large, which makes most existing reinforcement learning algorithms and strategies for solving MDP less effective. An initial study by authors has shown that a traditional algorithm, e.g., vanilla Q-learning, integrated with a neural network can only work for a small sized model, while cannot sufficiently address the model with a large space size [25].

Therefore, there is an urgent need to extend the research on microgrid technology from traditional residential sector, commercial sector, critical facilities, etc., to the manufacturing end-use customer, specifically, considering the joint energy control for both energy supply from the microgrid and energy load from the manufacturing system. In this paper, a joint control model conceding both microgrids and manufacturing systems is established using MDP. A novel reinforcement learning algorithm that leverages both on-policy temporal difference control (TD-control) and deterministic policy gradient algorithms (DPG) for continuous actions, together with function approximation of the action-value function via a neural network is proposed for solving the MDP. Experiments based on a manufacturing system with an onsite microgrid with renewable sources are implemented under real parameters to identify optimal control actions for both manufacturing system and microgrid components towards cost optimality. It is empirically validated that the optimal policies found by our reinforcement learning algorithm are more efficient in production and incur less cost when compared to randomly sampled policies and a routing operation policy.

The remaining part of the paper is organized as follows. Section II introduces the dynamic decision-making model using MDP. Section III introduces our novel neural-network integrated reinforcement learning algorithm that leverages both on-policy temporal difference control (TD-control) and deterministic policy gradient (DPG) algorithms. Section IV implements experiments on numerical case studies. Section V concludes the paper and discusses future works.

II. MARKOV DECISION PROCESS (MDP) FOR JOINT CONTROL OF MANUFACTURING AND ONSITE MICROGRID SYSTEMS

A. Formulate Joint Energy Control Problem Using Markov Decision Process (MDP)

A Markov Decision Process (MDP) model is proposed to model the decision-making of the joint control of both on-site microgrid generation system and manufacturing system. The microgrid system used is a typical setup consisting of a gas turbine generator, a battery bank as well as solar PV modules and wind turbines as shown in Figure 1. The manufacturing system modeled is a typical serial production line with \( N \) machines and \( N-1 \) buffers as shown in Figure 2 (where \( N = 5 \), and work-in-progress parts are stored in buffers). Let \( i = 1, 2, ..., N \) be the index of the machines and \( i = 1, 2, ..., N-1 \) be the index of the buffers.

The time horizon is discretized and divided into a set of discrete intervals, with the actual time duration for each interval to be \( \Delta t \). The time variable \( t \) denotes the indexes of the decision epochs of such discrete intervals at which the control actions identified based on the optimal policy and the given states can be implemented. The state, policy, state transition, objective function, and constraints of the proposed MDP are introduced as follows.

System State. Let the system states form a state space \( S \). The system state at decision epoch \( t \) is denoted by \( S_t \). It includes the states of manufacturing system \( (S_{mfg}^t) \), microgrid system \( (S_{mic}^t) \), and exogenous environmental features \( (S_{env}^t) \), which can be formulated by \( S_t = (S_{mfg}^t, S_{mic}^t, S_{env}^t) \). \( S_{mfg}^t \) can be denoted by \( S_{mfg}^t = (S_{mfg1}^t, ..., S_{mfgN_t}^t) \), where \( S_{mfg}(i) = 1, 2, ..., N \) denotes the state of machine \( i \) in the manufacturing system at decision epoch \( t \); \( S_{B}^t(i) = 1, 2, ..., N-1 \) denotes the state of the buffer \( i \) in the manufacturing system at decision epoch \( t \). Machine states include operational, blockage, starvation, off, and breakdown. Blockage means that the machine itself is not failed while the completed part cannot be delivered to the downstream buffer due to the breakdown of specific downstream machines. Starvation means that the machine itself is not failed while there is no incoming part from the upstream buffer due to the breakdown of specific upstream machines. The set of machine states is thus \{Opr, Blo, Sta, Off, Brk\}, where Opr, Blo, Sta, Off and Brk denote the operational state, blockage state, starvation state, off state, and breakdown state, respectively. At each state, there is a corresponding power consumption level, which can be illustrated by Figure 3. Buffer state is quantified by the number of work-in-process parts stored in each buffer at the decision epoch \( t \).

\( S_{mic}^t \) can be denoted by \( S_{mic}^t = (g_{t}^g, g_{t}^w, g_{t}^r, SOC_t) \), where \( g_{r}^r, g_{r}^w, \) and \( g_{r}^g \) denote the working status of solar PV, wind turbine, and generator, respectively, of the onsite microgrid generation system at decision epoch \( t \) (working= 1, not working= 0). A non-negative real number \( SOC_t \), denotes the
state of charge of the battery system at decision epoch \(t\).

The exogenous environmental feature state, \(S_{t}^{env}\), can be denoted by \(S_{t}^{env} = (I_t, v_t)\), where \(I_t\) denotes the solar irradiance at decision epoch \(t\), and \(v_t\) denotes the wind speed at decision epoch \(t\). The exogenous feature has an impact on the system dynamics and the cost function, but cannot be influenced by the control actions. The feature is time and weather dependent. The model formulation considers the availability of a deterministic forecast of the exogenous state information. The states \((I_t, v_t)\) are taken from one year’s data (assumed to be 360 days and 24 hours/day, so in total 8640 hours).

**Control Actions and Policy.** All admissible actions constitute an action space \(A \). Let \(\pi\) be the policy that maps from different states \(S\) to the actions \(A\). The control actions adopted at decision epoch \(t\) can be denoted by \(A_t\). It includes the control actions for the manufacturing system \((A_t^{mfg})\) and the microgrid system \((A_t^{mic})\), which can be denoted by \(A_t = (A_t^{mfg}, A_t^{mic})\). \(A_t^{mfg}\) can be denoted by \(A_t^{mfg} = (a_t^1, a_t^2, \ldots, a_t^N)\), where \(a_t^i (i = 1, 2, \ldots, N)\) is the control action for machine \(i\) at the decision epoch \(t\). The actions include K-action, W-action, and H-action. K-action intends to keep original machine states, which can be applied to the machines in \(\text{Opr, Blo, Sta, Off}\), and \(\text{Brk}\) states (note that machine repair is not considered a control action in this paper and repair is assumed to be a supposed-to-be reaction, so K-action used for breakdown machine can imply that repair will be implemented). H-action intends to turn off the machine, which can only be applied to the machine in \(\text{Opr, Blo}\), and \(\text{Sta}\) states. W-action intends to turn on the machine that was previously turned off, which can only be applied to the machine in \(\text{Off}\) states. Note that for model simplicity, we assume the energy consumption and time required for the transitions between different machine states can be ignored. \(A_t^{mic}\) specifies the actions with respect to the adjustment of the working status of the components of the microgrid as well as the corresponding energy flow and allocation in the joint system, which can be denoted by \(A_t^{mic} = (a_t^1, a_t^2, a_t^3, s_t^i, s_t^b, w_t^i, w_t^b, g_t^i, g_t^b, p_t^m, p_t^b, b_t^m)\). Here \(a_t^1, a_t^2, a_t^3\) are the actions of adjusting the working status (i.e., 1 is connected, 0 is not connected to the load) of the solar, wind, and generators in microgrids; \(s_t^i, s_t^b\), and \(s_t^b\) denote the solar energy used for supporting manufacturing, charging battery, and sold back to grids, respectively. Similarly, the notations \(w\) and \(g\) with corresponding superscripts denote the allocation of the energy generated by wind turbine and generator, respectively. \(p_t^m\) and \(p_t^b\) denote the use of the energy purchased from the grid, i.e., for supporting manufacturing, and charging battery, respectively. Note that the energy purchased from the grid is not considered for sold back. Finally, \(b_t^m\) denotes the energy discharged by the battery for supporting manufacturing, and is given by a binary variable \(\delta_t^{bm}\), so that \(b_t^m = b \cdot \delta_t^{bm} \cdot \Delta t\) for some discharging rate \(b > 0\). Note that the energy discharged by the battery is not considered for sold back.

**State Transition.** Let the function \(P: S \times A \times S \rightarrow [0, 1]\) be the transition probability function, so that \(P(S_t, A_t, S_{t+1}) = P_{R}(S_t, A_t, S_{t+1})\) is the probability of transition to state given that the previous state was \(S_t\) and action \(A_t\) was taken at state \(S_t\). The state transition given state \(S_t\) and adopted action \(A_t\) at decision epoch \(t\) is partially deterministic (that is, for some states \(S_t\) and \(A_t\) we have \(P(S_t, A_t, S_{t+1}) = 0\) or 1) and partially stochastic (that is, for some states \(S_t\) and \(S_{t+1}\) and actions \(A_t\) we have \(0 < P(S_t, A_t, S_{t+1}) < 1\), although they can all be put in the transition probability function \(P\). It is assumed that the state transition happens at the beginning of each interval when the decision is made.

For the manufacturing system, the buffer state at decision epoch \(t + 1\) can be obtained by (II.1) based on the states and the control actions adopted at decision epoch \(t\) of upstream and downstream machines:

\[
S_{t+1}^B = S_t^B + I(S_t^M, a_t^i) - I(S_t^M, a_t^i+1), \quad 0 \leq S_t^B \leq N_i, \quad (II.1)
\]

Here \(N_i\) is the capacity of buffer \(i\), and \(I(S_t^M, a_t^i)\) is an indicator function that is defined by (II.2):

\[
I(S_t^M, a_t^i) = \begin{cases} 
1, & \text{when } S_t^M = \text{Opr} \text{ and } a_t^i = K \\
0, & \text{when } S_t^M \neq \text{Opr} \text{ or } a_t^i = H.
\end{cases} \quad (II.2)
\]

Referring to the literature focusing on the statistical methods for machine reliability [26], we assume \(L_i\), which is the random lifetime of machine \(i\), follows Weibull distribution with specific shape parameter and scale parameter. The probability that machine \(i\) goes into breakdown or non-breakdown state at the next decision epoch \(t + 1\), given it is not in breakdown state at the current decision epoch \(t\) can be described by (II.3) and (II.4) respectively:

\[
Pr(S_{t+1}^M = \text{Brk}|S_t^M = \text{Brk}, S_t^M \neq \text{Off}) = Pr(L_i < t + \Delta t), \quad (II.3)
\]

\[
Pr(S_{t+1}^M = \text{Brk}|S_t^M = \text{Brk}, S_t^M \neq \text{Off}) = Pr(L_i \geq t + \Delta t). \quad (II.4)
\]

The probability whether the machine is at \(\text{Off}\) state or not at next decision epoch can be calculated by (II.5):

\[
S_{t+1}^M = \text{Off} \quad \text{if} \quad (S_t^M = \text{Off} \text{ and } a_t^i = K) \quad \text{or} \quad (S_t^M \neq \text{Off} \text{ and } a_t^i = H). \quad (II.5)
\]

In addition, we also assume \(D_i\), which is the random repair time of machine \(i\), follows Exponential distribution [27]. The probability that machine \(i\) completes or does not complete the repair at the next decision epoch \(t + 1\), given it is in repair at the current decision epoch \(t\) can be described by (II.6) and (II.7) respectively.

\[
Pr(S_{t+1}^M = \text{Brk}|S_t^M = \text{Brk}) = Pr(D_i < t + \Delta t), \quad (II.6)
\]

\[
Pr(S_{t+1}^M = \text{Brk}|S_t^M = \text{Brk}) = Pr(D_i \geq t + \Delta t). \quad (II.7)
\]
Thus, the probability that machine $i$ is in an Sta and Blo state can be described by (II.8) and (II.9), respectively:

$$
Pr(S_{i(t+1)}^M = \text{Sta}) = Pr(S_{i(t+1)}^M \neq \text{Brk or Off}) \cdot Pr(S_{i(t+1)}^B = 0) - Pr(S_{i(t+1)}^B = \text{Brk}) + Pr(S_{i(t+1)}^M \neq \text{Brk or Off}) \cdot Pr(S_{i(t+1)}^B = N_i) - Pr(S_{i(t+1)}^B = \text{Brk}) + Pr(S_{i(t+1)}^M \neq \text{Brk or Off}) \cdot Pr(S_{i(t+1)}^B = \text{Sta}) - Pr(S_{i(t+1)}^B = \text{Brk}) + Pr(S_{i(t+1)}^M \neq \text{Brk or Off}) \cdot Pr(S_{i(t+1)}^B = \text{Off}) - Pr(S_{i(t+1)}^B = \text{Brk}) .
$$

(II.8)

The probability that machine $i$ is in operation state can thus be calculated by (II.10):

$$
Pr(S_{i(t+1)}^M = \text{Opr}) = Pr(S_{i(t+1)}^M \neq \text{Brk}) - Pr(S_{i(t+1)}^M = \text{Sta}) - Pr(S_{i(t+1)}^M = \text{Blo}) .
$$

(II.10)

Therefore, the probability of system state operation transition between the current decision epoch and the next decision epoch can be calculated by using (II.1)-(II.10) when $A_t^m$ is adopted based on a given $S_t^m$.

For the microgrid system, the state transition of solar PV is determined by the action adopted. While the state transition of wind turbine is determined by the action adopted and the variation of the wind speed. They can be formulated by (II.11) and (II.12), respectively:

$$
g_{t+1}^s = \begin{cases} 
1, & \text{if } a_{t+1}^s = 1 \\
0, & \text{if } a_{t+1}^s = 0 
\end{cases},
$$

(II.11)

$$
g_{t+1}^w = \begin{cases} 
1, & \text{if } a_{t+1}^w = 1 \text{ and } v_{c1} \leq v_{t+1} \leq v_{c0} \\
0, & \text{if } a_{t+1}^w = 0 \text{ or } v_{t+1} > v_{c0} \text{ or } v_{t+1} < v_{c1} 
\end{cases},
$$

(II.12)

where $v_{c1}$ and $v_{c0}$ are the cut-in and cut-off wind speeds (m/s), respectively.

The state transition of control actions adopted, which can be formulated by (II.13):

$$
g_{t+1}^a = \begin{cases} 
1, & \text{if } a_{t+1}^a = 1 \\
0, & \text{if } a_{t+1}^a = 0 
\end{cases}.
$$

(II.13)

The state transition of battery (i.e., SOC) is determined by the charging and discharging happened between $t$ and $t+1$ as well as the original SOC, which can be formulated by (II.14).

$$
SOC_{t+1} = SOC_t + \left(s_t^1 + w_t^1 + g_t^1 + p_t^1\right)\eta - b_t^2/\eta,
$$

(II.14)

where $\eta$ is charging/discharging efficiency.

**Objective Function.** The objective is to identify an optimal policy based on the given state that can minimize the incurred cost from time $t$ to the end of decision horizon. The overall cost from state $S$ to state $S'$ under action $A$ is defined by $E(S,A,S') = E(S'|S,A)$. We will specify below that it is equal to energy consumption cost plus the microgrid operational cost, minus production throughput reward and the sold back reward. At decision epoch $t$, a transition from state $S_t$ to state $S_{t+1}$ under action $A_t$ results in an incurred cost $E(S_{t+1},A_t,S_t)$. The total incurred cost from time $0$ to the end of planning horizon, starting from state $S$ and under policy $\pi$, is given by

$$
C(S,\pi) = E \left[ \sum_{t=0}^{\infty} \gamma^t E(S_{t+1},\pi(S_t),S_t) \right] = E(S_0 = S).
$$

(II.15)

Here $\gamma \in [0,1)$ is the discount factor. The objective is to identify an optimal policy $\pi^* = \arg \min \{ C(S,\pi) \}$ that can guide the decision maker to find appropriate actions based on the given system state to minimize the total incurred cost $C(S,\pi)$ in (II.15).

For our model, we have in particular $E(S',A,S) = E(S,A)$ is the average total cost when action $A$ is taken at state $S$, which can be calculated by

$$
E(S,A) = TF(S,A) + MC(S,A) - TP(S,A) - SB(S,A),
$$

(II.16)

where $TF(S,A)$ is the cost for the energy purchased from the grid. $MC(S,A)$ is the operational cost for the onsite generation system. $TP(S,A)$ is the reward of production throughput of the manufacturing system, and $SB(S,A)$ is the sold back benefit. $TF(S,A)$ can be calculated by

$$
TF(S,A) = p_t \cdot r_t^i,
$$

(II.17)

where $p_t$ is the energy consumption purchased from the grid at decision epoch $t$. $r_t^i$ is the rate of energy consumption charge. $p_t$ can be calculated by

$$
p_t = E_t^{mfg} - (s_t^1 + w_t^1 + g_t^1 + b_t^2),
$$

(II.18)

where $E_t^{mfg}$ is the total energy consumed by the manufacturing system at decision epoch $t$ which can be determined by

$$
E_t^{mfg} = \sum_{i=1}^{N} PC_{it} \cdot \Delta t,
$$

(II.19)

where $PC_{it}$ is the amount of power drawn by the machine $i$ from $t$ to $t + 1$. $PC_{it}$ can be calculated by

$$
PC_{it} = \begin{cases} 
0, & \text{if } S_{it}^M = \text{Brk} \text{ or } S_{it}^M = \text{Off} \\
PC_{i}^{Opr}, & \text{if } S_{it}^M = \text{Opr} \\
PC_{i}^{Ind}, & \text{if } S_{it}^M = \text{Sta} \text{ or } S_{it}^M = \text{Blo}
\end{cases},
$$

(II.20)

where $PC_{i}^{Opr}$ and $PC_{i}^{Ind}$ are the power level of machine $i$ at the states of Opr and Sta/Blo, respectively.

$MC(S,A)$ can be calculated by

$$
MC(S,A) = c_t^s \cdot r_{omc}^s + c_t^w \cdot r_{omc}^w + c_t^g \cdot r_{omc}^g + \frac{2e(SOC_{max} - SOC_{min})}{r_{omc}^b} \cdot r_{omc}^b,
$$

(II.21)

where $c_t^s$, $c_t^w$, and $c_t^g$ are the energy generated at decision epoch $t$ from the onsite solar PV, wind turbine, and generator, respectively, which are calculated from the states and actions, that will be specified below. $r_{omc}^s$, $r_{omc}^w$, and $r_{omc}^g$ are the unit operational and maintenance cost for generating power from solar PV, wind turbine, and generator respectively. $r_{omc}^b$ is the operational and maintenance cost for battery storage.
system per unit charging/discharging cycle. $e$ is the capacity of battery storage system. $SOC_{max}$ and $SOC_{min}$ are maximum and minimum state of charge of battery storage system, respectively.

$e_t^r$ can be calculated according to [25]

$$e_t^r = \begin{cases} 0, & \text{if } g_t^w = 0, \\ I_t \cdot a \cdot \delta/1000, & \text{if } g_t^w = 1, \end{cases}$$

where $I_t$ is the solar irradiance of a certain location (W/m²) at decision epoch $t$, $a$ is the area of the solar PV system, and $\delta$ is the efficiency of the system.

$e_t^w$ can be calculated according to [18].

$$e_t^w = \begin{cases} 0, & \text{if } g_t^w = 0 \text{ or } v_t < v_{ci} \text{ or } v_t > v_{co}, \\ N_w \cdot R_{P_w} \cdot \Delta t, & \text{if } g_t^w = 1 \text{ and } v^r \leq v_t < v_{co}, \\ N_w \cdot R_{P_w} \cdot v_t - v_{ci}, & \text{if } g_t^w = 1 \text{ and } v_{ci} \leq v_t < v^r, \end{cases}$$

where $v_t$ is the wind speed (m/s) at decision epoch $t$, $v^r$ is the rated wind speeds (m/s), $N_w$ is the number of wind turbine in the onsite generation system and $R_{P_w}$ is the rated power of the wind turbine (kW). $R_{P_w}$ is determined by

$$R_{P_w} = \frac{1}{2} \rho \cdot \pi \cdot r^2 \cdot v_{avg}^3 \cdot \theta \cdot \eta_t \cdot \eta_g/1000,$$

where $\rho$ is the density of air, $v_{avg}$ is average wind speed, $\theta$ is the power coefficient, $r$ is the radius of the wind turbine blade, $\eta_t$ is its gearbox transmission efficiency, $\eta_g$ is electrical generator efficiency.

$e_t^g$ can be calculated by

$$e_t^g = \begin{cases} 0, & \text{if } g_t^g = 0, \\ n_g \cdot G_p \cdot \Delta t, & \text{if } g_t^g = 1, \end{cases}$$

where $n_g$ is the number of generators and $G_p$ is the rated output power of the generator (KW).

$TP(S, A)$ can be determined by

$$TP(S, A) = p_t \cdot r^p,$$

where $p_t$ is the production count at decision epoch $t$ and $r^p$ is the unit reward for each unit of production. $p_t$ can be calculated by

$$p_t = \begin{cases} 1, & \text{if } S_t^N = Opr \text{ and } a_t^N = K, \\ 0, & \text{if } S_t^N \neq Opr \text{ or } a_t^N = H. \end{cases}$$

Note that in this paper, the concern of reaching the target production throughput is represented as a monetary reward and integrated into the objective function. This strategy circumvents the challenges of modeling throughput as a major constraint in MDP. The throughput modeling and quantification for typical manufacturing systems with $N$ machines and $N - 1$ buffers as used in this paper are still major research challenges in the field of production system engineering when machine states of blockage & starvation are considered. Very limited research progresses have been made (see [28], [29]), however, these works cannot address the scenarios when dynamic control agents are involved.

Similarly, the sold back reward $SB(S, A)$ can be calculated by

$$SB(S, A) = s_t \cdot r^{sb},$$

where $s_t = s_t^b + w_t^b + g_t^b$ is the sold back energy to the grid at decision epoch $t$ and $r^{sb}$ is the unit reward from sold back energy.

### B. Parameterization of the Action Space and Constraints.

Our model contains the following constraints for the action space $A$, that are described below.

Since we set the battery state of charge level needs to be maintained within a given range, which can be formulated by

$$SOC_{min} \leq SOC_t + (s_t^b + w_t^b + g_t^b)\eta - \frac{b_{tm}^b}{\eta} \leq SOC_{max}.$$

Notice that (II.29) indicates that the actions to the microgrid on battery charging/discharging are controlled by the current SOC state.

Actions that can be applied to machines are restricted by the current machine states: K-action can be applied to the machine at $Opr, Blo, Sta, Off$ and $Brk$ states; H-action can only be applied to the machine at $Opr, Blo$ and $Sta$ states; W-action can only be applied to the machine at $Off$ states, i.e.,

$$a_t^i \in \{ K, H, W \}, \text{ if } S_t^N = Opr, Blo, Sta, Off, Brk ;$$
$$a_t^i \in \{ K, H \}, \text{ if } S_t^N = Opr, Blo, Sta ;$$
$$a_t^i \in \{ W \}, \text{ if } S_t^N = Off .$$

The energy flow balance for the energy generated by solar PV, wind turbine, and generator can be formulated by (II.31), (II.32), and (II.33), respectively.

$$s_t^m + s_t^b + s_t^g = e_t^g.$$

$$w_{t}^m + w_{t}^b + w_{t}^g = e_t^w.$$

$$g_{t}^m + g_{t}^b + g_{t}^g = e_t^g.$$

Notice that according to (II.22), (II.23) and (II.25), the energies $e_t^g, e_t^w, e_t^r$ depend on the working states of the microgrid $(g_t^g, g_t^w, g_t^r)$ and the solar irradiance $I_t$ and the wind speed $v_t$, the constraints (II.31)-(II.33) are restricting the actions applied to microgrid based on the current microgrid state and the environmental features.

The battery cannot be charged and discharged simultaneously. The charge/discharge constraint is represented as follows:

$$(s_t^b + w_t^b + g_t^b) \cdot b_{tm}^b = 0 .$$

Notice that since $b_{tm}^b = b_{tm}^b \cdot \cdot \cdot \cdot \Delta t$, we only seek for binary choices of $\delta_t^b = 0/1$ when $s_t^b = w_t^b = g_t^b = p_t^b = 0$.

The energy sold back to the grid and the energy purchased from the grid cannot happen simultaneously which can be represented by

$$(s_{t}^b + w_{t}^b + g_{t}^b)(r_{tm}^b + p_{t}^b) = 0 .$$

If the constraint (II.35) is satisfied at $s_{t}^b = w_{t}^b = g_{t}^b = 0$, so that we allow $p_{t}^m + p_{t}^b \neq 0$, then due to supply-demand
To simplify the model, we further assume that the energy purchased from the grid can be used either only for supporting manufacturing or charging battery, but not simultaneously, i.e.,

\[ p_t^m + p_t^b = 0. \tag{II.37} \]

Due to (II.37) and (II.35), if \( s_t^b = w_t^b = g_t^b = 0 \), we can introduce a binary variable \( \delta_t^{gb} = 0/1 \) (0 means purchased energy is not used for battery charging, 1 means purchased energy is used for battery charging) so that \( p_t^m = (1 - \delta_t^{gb})p_t 1_{p_t>0} \) and \( p_t^b = \delta_t^{gb} p_t 1_{p_t>0} \); if else, that is any of \( s_t^b, w_t^b \) or \( g_t^b \) is not equal to 0, we have \( p_t^m = p_t^b = 0 \).

To facilitate the design of policy-gradient related algorithms for training, we will further parameterize \((s_t^m, s_t^b, w_t^m, w_t^b, g_t^m, g_t^b, g_t^{gb})\) by introducing proportionality parameters

\[ \theta = (\lambda^m, \lambda^b, \lambda_w^m, \lambda_w^b, \lambda_g^m, \lambda_g^b) \tag{II.38} \]

and the representation

\[
\begin{align*}
\{s_t^m &= e_t^m \cdot \lambda^m_s, s_t^b = e_t^b \cdot \lambda^b_s, s_t^{gb} = e_t^{gb} \cdot (1 - \lambda^m_s - \lambda^b_s), \\
\{w_t^m &= e_t^m \cdot \lambda^m_w, w_t^b = e_t^b \cdot \lambda^b_w, w_t^{gb} = e_t^{gb} \cdot (1 - \lambda^m_w - \lambda^b_w), \\
\{g_t^m &= e_t^m \cdot \lambda^m_g, g_t^b = e_t^b \cdot \lambda^b_g, g_t^{gb} = e_t^{gb} \cdot (1 - \lambda^m_g - \lambda^b_g). \tag{II.39} \}
\end{align*}
\]

These representations further simplify the constraints (II.31)-(II.33) into the following constraints

\[
\begin{align*}
\lambda^m_s &\geq 0, \lambda^b_s \geq 0, 0 \leq \lambda^m_w + \lambda^b_w \leq 1, \tag{II.40} \\
\lambda^m_w &\geq 0, \lambda^b_w \geq 0, 0 \leq \lambda^m_g + \lambda^b_g \leq 1, \tag{II.41} \\
\lambda^m_g &\geq 0, \lambda^b_g \geq 0, 0 \leq \lambda^m_\theta + \lambda^b_\theta \leq 1. \tag{II.42} 
\end{align*}
\]

To further deal with the constraints (II.34), (II.35) and (II.37), we further introduce the binary (0/1) variables \( \delta_t^{gb} = 1_{s_t^b + w_t^b + g_t^b > 0} \) and \( \delta_t^g = 1_{p_t^g > 0} \). Then constraints (II.34), (II.35) and (II.37) become a discrete constraint

\[
(\delta_t, \delta_t^g, \delta_t^{gb}, \delta_t^g) \in \{(1, 1, 1, 0, 0), (1, 1, 0, 0, 0), (1, 0, 0, 1), (0, 1, 1, 0, 0), (0, 0, 0, 1, 1) \}. \tag{II.43}
\]

Notice that (II.43) summarizes all discrete constraints for the control parameters on the microgrid. The remaining continuous constraints for the microgrid are only (II.29) and (II.40)-(II.42).

Based on (II.43), we can further write the constraints (II.29) and (II.40)-(II.42) into different constraints on \( \theta \) (the variable in (II.38)):

\[
(\delta_t^b, \delta_t^g, \delta_t^{gb}, \delta_t^{gb}, \delta_t^{gbm}) = (0, 0, 0, 1, 1). \]

The constraints on \( \theta \) are given by

\[
\begin{align*}
\lambda^b_s &= \lambda^b_w = \lambda^b_g = 0; \\
0 &\leq \lambda^m_s, \lambda^m_w, \lambda^m_g < 1; \\
SOC_{min} &\leq SOC_t - b \Delta t/\eta \leq SOC_{max}; \tag{II.44}
\end{align*}
\]

(2) \( (\delta_t^b, \delta_t^g, \delta_t^{gb}, \delta_t^{gb}, \delta_t^{gbm}) = (0, 1, 0, 0, 1) \). The constraints on \( \theta \) are given by

\[
\begin{align*}
\lambda^b_s &= \lambda^b_w = \lambda^b_g = 0; \\
\lambda^m_s &= \lambda^m_w = \lambda^m_g = 1; \\
SOC_{min} &\leq SOC_t - b \Delta t/\eta \leq SOC_{max}; \tag{II.45}
\end{align*}
\]

(3) \( (\delta_t^b, \delta_t^g, \delta_t^{gb}, \delta_t^{gb}, \delta_t^{gbm}) = (0, 1, 1, 0, 0) \). The constraints on \( \theta \) are given by

\[
\begin{align*}
\lambda^b_s &= \lambda^b_w = \lambda^b_g = 0; \\
\lambda^m_s &= \lambda^m_w = \lambda^m_g = 1; \\
SOC_{min} &\leq SOC_t + \eta (E_{mfg} - (s_t + w_t^m + g_t^m + b \Delta t)) \\
&\cdot 1_{(E_{mfg} - (s_t + w_t^m + g_t^m + b \Delta t)) > 0} \leq SOC_{max}. \tag{II.46}
\end{align*}
\]

(4) \( (\delta_t^b, \delta_t^g, \delta_t^{gb}, \delta_t^{gb}, \delta_t^{gbm}) = (1, 0, 0, 0, 0) \). The constraints on \( \theta \) are given by

\[
\begin{align*}
\lambda^b_s &\geq 0, \lambda^b_w > 0, \lambda^b_s + \lambda^b_w = 1; \\
\lambda^m_s &\geq 0, \lambda^m_w > 0, \lambda^m_s + \lambda^m_w = 1; \\
SOC_{min} &\leq SOC_t + \eta (s_t + w_t^b + g_t^b) \leq SOC_{max}. \tag{II.47}
\end{align*}
\]

(5) \( (\delta_t^b, \delta_t^g, \delta_t^{gb}, \delta_t^{gb}, \delta_t^{gbm}) = (1, 0, 1, 0, 0) \). The constraints on \( \theta \) are given by

\[
\begin{align*}
\lambda^b_s &\geq 0, \lambda^b_w > 0, 0 \leq \lambda^b_s + \lambda^b_w < 1; \\
\lambda^m_s &\geq 0, \lambda^m_w > 0, 0 \leq \lambda^m_s + \lambda^m_w < 1; \\
SOC_{min} &\leq SOC_t + \eta (s_t + w_t^b + g_t^b) \leq SOC_{max}. \tag{II.48}
\end{align*}
\]

(6) \( (\delta_t^b, \delta_t^g, \delta_t^{gb}, \delta_t^{gb}, \delta_t^{gbm}) = (1, 1, 1, 0, 0) \). The constraints on \( \theta \) are given by

\[
\begin{align*}
\lambda^b_s &\geq 0, \lambda^b_w > 0, \lambda^b_s + \lambda^b_w = 1; \\
\lambda^m_s &\geq 0, \lambda^m_w > 0, \lambda^m_s + \lambda^m_w = 1; \\
SOC_{min} &\leq SOC_t + \eta (s_t + w_t^b + g_t^b) \\
&\cdot 1_{(E_{mfg} - (s_t + w_t^m + g_t^m + b \Delta t)) > 0} \leq SOC_{max}. \tag{II.49}
\end{align*}
\]

All effective constraints for the admissible actions in this problem are (II.30), (II.43) and (II.44)-(II.49).

### III. NOVEL NEURAL-NETWORK INTEGRATED REINFORCEMENT LEARNING ALGORITHMS FOR THE MDP MODEL

#### A. Review of Previous Works and Our Contributions

As far as the authors are aware of, there are very few published works that address the joint control of manufacturing and onsite microgrid system using MDP and reinforcement learning algorithms. A few existing works that are in this direction are [30], [31], where Deep Q-learning (DQN) algorithms have been applied to the learning of the microgrid system only.

**Our Contributions.**

1. We have designed a new model that combines joint control of manufacturing and onsite microgrid system.
(2) Our novel reinforcement learning algorithm integrates deterministic policy-gradient (DPG) with on policy temporal difference (TD) control to treat the co-existence of discrete and continuous states and actions. We also address the constraints via proximal projection operators and policy gradient updates.

B. Abstract Formulation of the Model, Deep Reinforcement Learning Algorithm for solving the Model in its Abstract Formulation

A state $S_t \in \mathcal{S}$ in the space of the model consists of two parts $S_t = (S^d_t, S^c_t)$: the discrete part

$$S^d_t = (S^{M}_{1t}, S^{M}_{2t}, ..., S^{M}_{Nt}, S^{B}_{1t}, S^{B}_{2t}, ..., S^{B}_{(N-1)t}, g^{I}_{t}, g^{w}_{t}, g^{Q}_{t}, I_t, v_t)$$

(III.1)

which consists of the machine, buffer and microgrid states, as well as the coarse-grained solar irradiance and the wind speed $(I_t, v_t)$. Here to reduce complexity, an approximate course-grain scheme will be applied to each pair of the values $(I_t, v_t)$, so that they will be approximated by integers closest to them on a grid with $20 \times 20$ states, thus taken values among $20 \times 20$ different states; the continuous part

$$S^c_t = (SOC_t),$$

(III.2)

which consists of the SOC state.

An action $A_t$ in the action space of the model also consists of three parts $A_t = (A^d_t, A^c_t, A^f_t)$: the discrete part

$$A^d_t = (a^1_t, a^2_t, ..., a^N_t, a^w_t, a^Q_t, \theta^b_t, \delta^b_t, \delta^{lb}_t, \delta^{lb}_w)$$

(III.3)

which consists of the actions on each of the machines and the connected/disconnected action of the solar PV, wind turbine and the generator, as well as the indicator variables in (II.43); the continuous part

$$A^c_t = (s^b_t, s^b_{w}, s^b_{m}, w^m_t, w^b_t, w^b_{m_t}, g^m_t, g^b_t, g^b_{m_t})$$

(III.4)

which consists of the solar, wind, generator energy used for supporting manufacturing, charging battery and sold back to the grid. The continuous part will be parameterized by the variable $\theta$ introduced in (II.38), so that we have $A^c_t = A^c(\theta_t, S_t)$; the remainder part $A^f_t = (p^m_t, p^b_t, b^m_t)$, which consists of the use of the energy purchased from the grid for supporting manufacturing and charging battery, and the energy discharged by the battery for supporting manufacturing. These can be calculated directly from $\delta^{lb}$ and $\delta^{lbw}$ as well as (II.36), (II.18), which then can be calculated from $A^c_t$.

At a specific state $S_t \in \mathcal{S}$, the actions that can be taken are restricted by this particular state via the restriction

$$A^d_t \in D^d(S^d_t),$$

(III.5)

and

$$\theta \in D^c(S^c_t, S^d_t, A^d_t),$$

(III.6)

where $D^d$ is the set of admissible discrete actions $A^d_t$ that can be taken at the current state, and $D^c$ is the set of admissible parameters $\theta$ for continuous actions $A^c_t$ that can be taken at the current state. According to the discussions in section II-B, we see that $D^d$ is given by (II.30) and (II.43), and depends only on the discrete part of the current state (actually, only on the current states of the machines) and $D^c$ is given by one of (II.44)-(II.49), that depend on both the continuous and the discrete parts of the current state, as well as the discrete actions.

From the above abstract formulation, we see the mathematical complicity of the problem in that the state and action spaces contain both discrete and continuous parts, and the action constraints are determined by both the discrete and continuous parts of the states, as well as the discrete part of the actions. To design an effective learning algorithm, we propose to integrate both the on-policy TD control (SARSA) for finding the discrete part of the optimal control actions and the proximal projection of the deterministic policy gradient method associated with on-policy actor-critic (see [32]) for finding the continuous part of the optimal control actions. We employ on-policy methods rather than off-policy to deal with constraints that are variable with respect to change of states. This enables more exploration over the variable constraints. In the use of SARSA and actor-critic, we borrow the ideas in Deep-Q Learning (see [33], [34], [35]) and we use a neural network to serve as the function approximator of the action-value function $Q(S,A)$. Proximal algorithm (see [36]) is a popular optimization technique in machine learning for handling constrained optimization problems, and here we combine it with the deterministic policy gradient iterations to approximate the continuous part of the optimal control policies. We write the proposed algorithm into pseudo-code at Algorithm 1.

C. Solution Algorithm for the Original Model

One major challenge of solving the MDP problem using Algorithm 1 lies in that the set of admissible parameters $\theta \in D^c(S,A)$ for continuous actions $A^c$, that are determined by (II.44)-(II.49), have an intersection structure. Indeed from (II.44)-(II.49) one can view $D^c(S,A)$ as an intersection

$$D^c(S,A) = D^c \cap D^{SOC}(\theta, SOC, (\delta^{lb}, \delta^{lbw}, \delta^{lb}_w, \delta^{lbw}_w))$$

Here we set the fixed simplex

$$D^c = \{(\lambda^m_w, \lambda^b_w, \lambda^b_{wm}, \lambda^b_{wmb}) \in [0, \infty)^4 : \lambda^m_w, \lambda^b_w, \lambda^b_{wm}, \lambda^b_{wmb} \geq 0, 0 \leq \lambda^m_w + \lambda^b_w \leq 1, 0 \leq \lambda^b_w + \lambda^b_{wm} \leq 1, 0 \leq \lambda^b_{wmb} \leq 1\}$$

(III.7)

The complicacy in $D^c(S,A)$ lies in the other part of the intersection $D^{SOC}(\theta, SOC, (\delta^{lb}, \delta^{lbw}, \delta^{lb}_w, \delta^{lbw}_w))$, that may vary according to the choice of $(\delta^{lb}, \delta^{lbw}, \delta^{lb}_w, \delta^{lbw}_w)$ and depend on the SOC state, which makes the proximal projection to $D^c(S,A)$ in Step 7 of Algorithm 1 nearly impossible to compute in practice.

To fix this issue, we suggest to relax the constraint $D^c(S,A)$ by only considering its fixed simplex part $D^c$. Of course, if we only project to $D^c$ at every proximal projection step in our Step 7 of Algorithm 1, we may miss the SOC constraints in (II.44)-(II.49). But we can then fix this issue by using the $\theta$ found on the relaxed constraint set $D^c$ and determine the binary variables $(\delta^{lb}_1, \delta^{lbw}_1, \delta^{lb}_w, \delta^{lbw}_w)$, as well as update the SOC. If the SOC values we obtain violate the additional constraints in $D^{SOC}(\theta, SOC_1, (\delta^{lb}, \delta^{lbw}, \delta^{lb}_w, \delta^{lbw}_w))$, we will just set them to be the boundary values $SOC_{max}$ or $SOC_{min}$, so that they will not violate the SOC constraints.
**Algorithm 1** Training the Abstract Model via Integrating on-policy TD control (SARSA) and Proximal Projection of the Deterministic Policy Gradient

1: **Input:** Input state space $S$, action space $A$, constraints $D^d$, $D^r$; Given the neural network architecture $Q(S, A^d, A^ε(θ); ω)$; Discount factor $0 < γ < 1$; Learning rates $ηθ, ηω > 0$;

2: **Initialization:** Initialize the weight vector $w_0 \propto$ a given prior distribution; initial action $A^d_0, A^ε_0$ and action parameter $θ_0$, $A^ε_0 = A^ε(θ_0)$; Initial state $S_0 = (S^d_0, S^r_0)$;

3: for $t = 0, 1, 2, …$ do

4: Run one step of the MDP from state $S_t$ under action $A_t = (A^d_t, A^ε_t)$, obtain a new state $S_{t+1}$;

5: Calculate the total cost $E(S_t, A_t)$;

6: Identify

$$A^d_{t+1} = \arg \min_{A^d \in D^d(S_{t+1})} Q(S_{t+1}, A^d, A^ε(θ_t); ω_t)$$

7: Based on $A^d_{t+1}$, update the policy parameter $θ$ according to deterministic policy gradient

$$θ_{t+1} = \text{prox}_{D^r(S_{t+1}, A^ε_{t+1})} [θ_t - 1/ηθ \nabla_θ Q(S_t, A^d_t, A^ε_t; ω_t)]$$

8: Based on $A^d_{t+1}$ and $θ_{t+1}$, obtain $A^ε_{t+1}$, so that

$$A_{t+1} = (A^d_{t+1}, A^ε_{t+1}) = A^ε(θ_{t+1}), A^ε_{t+1})$$

9: Calculate on policy TD:

$$δ_t = E(S_t, A_t) + γ Q(S_{t+1}, A_{t+1}; ω_t) - Q(S_t, A_t; ω_t)$$

10: Update the weight vector $ω$ using actor-critic:

$$ω_{t+1} = ω_t - ηω δ_t ω Q(S_t, A_t; ω_t)$$

11: end for

12: **Output:** With the given optimal $ω^*$ and $θ^*$ found, for each state $S$ given, output the approximate optimal policy $(A^*, A^{ε^*}(θ^*), A^{r*})$ where

$$Q(S, A^*, A^{ε^*}(θ^*), A^{r*}) = \arg \min_{A^d, A^ε, A^r \text{admissible}} Q(S, A^d, A^ε(θ), A^r; ω*)$$

In this section, numerical case studies are implemented to illustrate the benefits of the proposed modeling and solution strategies. Our source code is open and can be found at the GitHub repository [37] for this project.

We have carried out experiments of the manufacturing-microgrid system using a real-case parameter set. The manufacturing system includes five machines and four buffers as shown in Figure 2. The parameters of the manufacturing system are taken according to [38]. Specifically, the parameters related to machines and buffers of the manufacturing system are shown in Table I and Table II, respectively. Note that the mean time between failures of each machine is modeled as random variables follow Weibull distribution with respective scale and shape parameters. The mean time to repair of each machine is modeled as exponentially distributed random variables. Unit production reward $r_p$ is set to be $10^3$ per unit.

The parameters of the microgrid used in the experiment are sized based on the manufacturing load according to the methods in [18]. The parameters related to wind turbine, battery storage system, and solar panel and generator are illustrated in Table III, Table IV, and Table V, respectively. The data of solar irradiance and wind speed are collected from Solar Energy Local [39] and State Climatologist of Illinois [40], respectively.

In order to prevent computational overflow, in our actual numerical experiment we scaled all the parameters by choosing different units of measurement, with distance measured by km $(10^3)$ m, time measured by hour $(60 \text{ min}= 3600)$s, speed measured by km/h $(3.6 \text{ m/s})$, energy measured by MegaWatt $(10^6 \text{ W})$, money cost measured by $10^4$ US$, area measured by km$^2 $(10^6 \text{ m}^2)$, mass measured by $10^6 \text{ kg}$. Time period is illustrated in hours. The neural network $Q(S, A; ω)$ that we use to simulate the action-value function $Q(S, A)$ contains two hidden layers with 100 neurons for each layer, with Sigmoid and ReLU activations for layers 1 and 2. The output is then scaled back to usual units of measurement, like $\$ for cost and kW for energy demand.
We take the discount factor $\gamma = 0.999$. The experiment includes reinforcement learning training for $5 \times 10^3$ iterations, with each iteration counts for time period with equal duration of one hour per period, and the learning rates are tuned to be $\eta_0 = 0.003$ and $\eta_{\omega} = 0.0003$, with $\eta_{\omega}$ discounted by a factor 0.999 at each iteration. Figure 4-(a) plots the $L^2$-norm of the difference at two consecutive iterations during training in the neural network weights ($\|w_{t+1} - w_t\|_2^2$), which clearly indicates the convergence of the training process. Figure 4-(b) plots the cumulative reward function (total incurred cost $C(S, \pi)$ in (II.15) truncated at the current iteration step).

In order to validate the effective convergence of our reinforcement learning algorithm, comparison has been carried out for a pure Q-learning algorithm for the same microgrid-manufacturing system with a smaller size (2 machines and 1 buffer) [25]. After discretization of the continuous states and actions, the state space has a size of $3.8 \times 10^4$ and the action space has a size of $2.6 \times 10^4$. The action-value function ($Q$-function) is approximated by a smaller fully-connected neural network with two hidden layers and Sigmoid activation, where each hidden layer has 32 neurons. Again, we calculated the square norms of the differences of the neural-network weight vectors for each two consecutive algorithm iterations (i.e. $\|w_{t+1} - w_t\|_2^2$). The discount factor $\gamma = 0.1$. The neural network is trained using Adam [41] with different learning rates: Figure 4-(c) is for learning rate 0.001 and Figure 4-(d) is for learning rate 0.0001. It is seen that in these two cases, even after $10^4$ iterations, the pure Q-learning algorithm cannot converge due to the immense size of the discretized state-action spaces (a manifestation of the “curse of dimensionality”), indicating the effectiveness of our method that combines deterministic policy gradient with discrete Monte-Carlo type searches.

Based on the optimal parameters ($\omega^\ast$, $\theta^\ast$) found for the neural-network $Q(S, A; \omega^\ast)$ and the continuous action $A^\ast(\theta^\ast)$, we tested the corresponding microgrid-manufacturing model at a time horizon of 100 time periods, with each time period equals one hour. At each decision point with state $S$, we store all admissible actions $A_d^\ast$ and remainder actions $A^\ast$ in a tree and search over the tree to identify optimal actions $(A_d^\ast, A^\ast)$, i.e.,

$$A^\ast = \arg \max Q(S, A_d^\ast, A^\ast(\theta^\ast), A^\ast; \omega^\ast).$$

These optimal actions are then implemented for the MDP system to jump to the next state and the incurred cost, throughput and energy demand are calculated. The results are compared with two baseline scenarios: The first one runs under a random policy, while the second one is a routing policy. For the random policy, the accumulated total incurred cost $E(S, A)$ at (II.16), total energy cost $TF(S, A)$ at (II.17) and total production units $p_t$ in (II.26) for the optimal policy and random policy are calculated for the system running at a total horizon of 100 time periods (each period = 1 hour). The system under randomly chosen policy starts with the same initial conditions as the system under optimal policy. The results of 3 experiments are shown in Figure 5, where red solid lines are for the optimal policy selected by reinforcement learning and black star lines are for the random policy. It is clearly seen from these results that under the optimal policy the manufacturing system tends to produce more throughput with less total cost and similar or less total energy cost (energy demand). More precisely, in one of the experiments, the optimal policy found by reinforcement learning over a time horizon of 100 time periods has an output (the quantity $p_i$ in (II.26)) of 73, while the randomly selected policy only produces 24. At the same time, the total cost for the optimal policy is $-728293.1002207897$ and the total cost for the random policy is $-236214.01230431726$. In terms of energy cost, optimal policy is also about one time less than the random policy, with $876.4532621691491$ for optimal policy and $1656.0785448269392$ for random policy. The two other experiments behave very similarly.

For the routing policy, we consider a routing practice strategy that can be adopted by many industrial practitioners, i.e., the production system and microgrid are controlled or scheduled separately. The production scheduling is generated to minimize total energy consumption without sacrificing target production. This model is briefly introduced as follows. Let $x_{it}$ be the binary decision variable denoting the production schedule of the manufacturing system, i.e., it takes the value of one when machine $i$ is scheduled for production in period $t$, and zero otherwise. The objective function can be formulated as:

$$\min_{x_{it}} \sum_{i \in T} x_{it} \cdot p_i \cdot \Delta t,$$

where $T$ is the set including all time periods $t$, $p_i$ is the

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C (\text{m}^2)$-Panel area</td>
<td>1400</td>
<td>$n_p$ (unit)-Number of generator</td>
<td>1</td>
</tr>
<tr>
<td>$\theta$-Efficiency</td>
<td>0.2</td>
<td>$G_p$ (kW)-Generator capacity</td>
<td>650</td>
</tr>
<tr>
<td>$r_{\text{omc}}$ (S/KWh)</td>
<td>0.17</td>
<td>$r_{\text{omc}}$ (S/KWh)-Operating cost</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table IV: Battery Storage Parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C (\text{m}^2)$-Panel area</td>
<td>1400</td>
<td>$n_p$ (unit)-Number of generator</td>
<td>1</td>
</tr>
<tr>
<td>$\theta$-Efficiency</td>
<td>0.2</td>
<td>$G_p$ (kW)-Generator capacity</td>
<td>650</td>
</tr>
<tr>
<td>$r_{\text{omc}}$ (S/KWh)</td>
<td>0.17</td>
<td>$r_{\text{omc}}$ (S/KWh)-Operating cost</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table V: Solar Panel and Generator Parameters.
rated power of machine $i$, $\Delta t$ is the time duration of each discretization period. Note that for simplicity, the values of $PC_{i}^{Opr}$ are used to for the rated power without considering the difference between $PC_{i}^{Opr}$ and $PC_{i}^{Idl}$.

Two constraints are formulated as follows:

$$\sum_{t\in T} x_{Ni} \cdot PR_{N} \geq TA \quad \text{(IV.2)}$$

where $x_{Ni}$ is the decision variable for machine $N$ (i.e., the last machine) of the production system. $PR_{N}$ is the production rate of machine $N$. $TA$ is the target production count. This constraint shows that the target production should be satisfied. Note that this constraint is based on a simplified assumption that machine breakdown and the resultant blockage/starvation are not considered.

$$0 \leq B_{i(t+1)} = B_{it} + x_{it} \cdot PR_{it} - x_{(i+1)t} \cdot PR_{i+1} \leq C_{i} \quad \text{ (IV.3)}$$

where $C_{i}$ is the capacity of buffer $i$, $B_{i(t+1)}$ is the count of work-in-progress parts stored in buffer location $i$ at the beginning of period $t+1$. This constraint shows material flow balance and the work-in-progress part in each buffer location cannot exceed its respective bounds.

After solving the aforementioned Integer Programming, the production schedule that can minimize energy consumption without sacrificing production can be obtained. Then, the utilization of microgrids follows the following empirical rules will be implemented. First, the battery storage system and generator are typically considered backups for emergency situations in practice, and thus are not used in this routing policy.

Second, if the renewable sources are available at time period $t$, they will be first used to satisfy the energy demand of production. If the renewable sources have a higher supply capability than production demand at period $t$, the remaining part will be sold back to the grid. If the renewable sources have a less supply capability than production demand at period $t$, the demand gap will be filled by purchasing electricity from the grid. The wind energy has a priority to be used as the solar source since wind energy cost is typically lower than solar energy cost.

To match the target production unit made by optimal policy, for this routing strategy found by mixed-integer programming, we set the target output at the time horizon 100, i.e., the target production unit $TA$ in (IV.2) to be equal to 73, which is the total production throughput in units for the optimal policy (the quantity $p_t$ in (II.26)) at time horizon 100. We found that the total energy cost under the optimal policy found by this integer programming is $3642.8834007673$. This has been about four times of our previously announced $876.4352621091491$ for the optimal policy found by reinforcement learning. Two other experiments are made and the results are similar. The results of the evolution of the total energy cost and total production throughput in units as a function of time are shown in Figure 5, where red solid lines are for the optimal policy selected by reinforcement learning and blue dashed lines are for the routing strategy found by mixed-integer programming. It is clearly seen that the reinforcement learning selects a policy that incurs less energy cost.

The overall comparison between proposed reinforcement learning model and random policy as well as routing policy is summarized in Table VI.

![Figure 5: Left to Right: Average over 3 experiments the comparison of (a) total throughput in production unit; (b) total energy cost incurred by optimal, routing and random policies: red solid line = optimal policy, blue dashed line = routing strategy via mixed-integer programming, black star line = random policy.](image)

### V. CONCLUSION

This paper proposes a joint dynamic control model of microgrids and manufacturing systems using Markov Decision Process (MDP) to identify an optimal control strategy for both microgrid components and manufacturing system so that the energy cost for production can be minimized without sacrificing production throughput. A novel reinforcement learning algorithm that leverages both on-policy temporal difference control (TD-control) and deterministic policy gradient (DPG) algorithms is proposed to resolve the joint control of microgrid and manufacturing system. Experiments for a manufacturing system with an onsite microgrid with renewable sources have been implemented and the results show the effectiveness of combining TD control and policy gradient methodologies in addressing the “curse of dimensionality” in dynamic decision-making with high dimensional and complicated state and action spaces.

For future work, real time decision making can be considered for emergency situations such as natural disasters that lead to non-availability of external energy supplies from grids.

### APPENDIX A

**THE PROJECTION ONTO THE SIMPLEX $D^c$**

In this appendix we calculate directly the projection operator prox$_{D^c}(\theta)$ for a given $\theta \in \mathbb{R}^6$ onto the simplex

$$D^c = D^c_v \times D^c_w \times D^c_g ,$$

where $D^c_v$, $D^c_w$, $D^c_g$ are three simplices all isomorphic to

$$D = \{(\lambda_1, \lambda_2) : \lambda_1 \geq 0, \lambda_2 \geq 0, 0 \leq \lambda_1 + \lambda_2 \leq 1\} .$$
Let \( \theta = (\theta_s, \theta_w, \theta_g) \) where each \( \theta_s, \theta_w, \theta_g \in \mathbb{R}^2 \). Then we have

\[
\text{prox}_{D^\theta} (\theta) = (\text{prox}_{D^\theta_s} (\theta_s), \text{prox}_{D^\theta_w} (\theta_w), \text{prox}_{D^\theta_g} (\theta_g)).
\]

If \( \theta_0 \in D \), then \( \text{prox}_{D^\theta} (\theta_0) = \theta_0 \). For each \( \theta_0 = (\theta_s^0, \theta_w^0, \theta_g^0) \in \mathbb{R}^2 \cdot D \), we can easily calculate \( \text{prox}_{D^\theta} (\theta_0) \) according to the following rules:

(I) \( \theta_s^0 < 0, \theta_w^0 < 0 \), then \( \text{prox}_{D^\theta} (\theta_s^0, \theta_w^0) = (0, 0) \);

(II) \( 0 \leq \theta_s^0 \leq 1, \theta_w^0 < 0 \), then \( \text{prox}_{D^\theta} (\theta_s^0, \theta_w^0) = (\theta_s^0, 0) \);

(III) \( \theta_s^0 > 1 \) and \( \theta_w^0 < 0 \), \( \theta_s^0, \theta_w^0 > 0, \theta_s^0 > 1 \) and \( -1 \leq \theta_w^0 - \theta_w^0 \leq 1 \), then

\[
\text{prox}_{D^\theta} (\theta_s^0, \theta_w^0) = \left( \frac{1 + \theta_s^0 - \theta_w^0, 1 - \theta_s^0 + \theta_w^0}{2}, \frac{1}{2} \right);
\]

(IV) \( \theta_s^0 > 1 \) and \( \theta_w^0 < -1 \), then \( \text{prox}_{D^\theta} (\theta_s^0, \theta_w^0) = (0, 1) \);

(V) \( 0 \leq \theta_s^0 \leq 1, \theta_w^0 < 0 \), then \( \text{prox}_{D^\theta} (\theta_s^0, \theta_w^0) = (0, \theta_w^0) \).

\[ \text{REFERENCES} \]


[37] Source code for our paper: https://github.com/huwening0606/r-m-manufacturing.


Wenqing Hu received the B.Sc. degree in Applied Mathematics from Peking University, Beijing, China, in 2008, the Ph.D. degree in Mathematics from the University of Maryland, College Park, MD, USA, in 2013. From 2016 to now he is serving as an Assistant Professor in Mathematics at the Department of Mathematics and Statistics, Missouri University of Science and Technology, Rolla, MO, USA.

His research interests lie in probability theory and statistical methodology. He has been working on problems in data sciences, statistical machine learning and optimization.

Zeyi Sun received the B.Eng. degree in material science and engineering from Tongji University, Shanghai, China, in 2002, the M.Eng. degree in manufacturing from the University of Michigan Ann Arbor, Ann Arbor, MI, USA, in 2010, and the Ph.D. degree in industrial engineering and operations research from the University of Illinois at Chicago, Chicago, IL, USA, in 2015. He served as an Assistant Professor with the Department of Engineering Management and Systems Engineering, Missouri University of Science and Technology, Rolla, MO, USA, from 2015 to 2020. Currently, he is a senior research scientist with Mininglamp Academy of Sciences, Mininglamp Technology, Beijing, China.

His research interest is mainly focused on using reinforcement learning algorithms to solve dynamic decision-making problem formulated by Markov Decision Process.

Jiaojiao Yang received the B.Sc. degree in Chaobu College, Hefei, Anhui, China in 2011, and the Ph.D. degree in Applied Mathematics from the South China University of Technology in 2016. From 2016 to 2018 she is serving as a Researcher at Huazhong University of Science and Technology, Wuhan, China, and since 2018 to now she has been an Assistant and Associate Professor at Anhui Normal University, Wuhu, Anhui, China.

Her research interests are in the analysis of fractals, dynamical systems and machine learning.

Louis Steimleiser received his M.Sc. in Applied Mathematics from the Missouri University of Science and Technology through a joint program with Ulm University in Germany. The emphasis was in Mathematical Statistics and Machine Learning as well as their applications to Finance. During his time in Ulm he started his own consulting business after his experience in Risk Management at KPMG. In 2016, he was awarded his B.Sc. in Business Mathematics from the University of Hambach, having emphasized in Finance, Stochastic Processes, and Statistics. Louis’ research interests revolve around algorithmic trading, reinforcement learning, machine learning, financial mathematics, and mathematical statistics.

Kaibo Xu Dr. Kaibo Xu received his Bachelor degree (1998) in Computer Science from Beijing University of Chemical Technology and his Master (2005) and PhD (2010) in Computer Science from the University of the West of Scotland. He worked as a Teaching Assistant (1998-2004), Lecturer (2004-2009), Associate Professor (2009-2017) at Beijing Union University. He has supervised more than 20 master and doctoral students who are successful in their academic and industrial careers. As the principal investigator, he has received 7 governmental funds and 5 industrial funds with the total amount of 5M in the Chinese dollar. Dr. Kaibo Xu has also consulted extensively and been involved in many industrial projects. He worked as the Chief-Information-Officer (CIO) of Yunbai Clothing Retail Group, China (2016-2019). Currently, he is serving as the vice president and principal scientist of MiningLamp Tech. His research interests include graph mining, knowledge graph and knowledge reasoning.