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

Jiaojiao Yang, Zeyi Sun*, Wenqing Hu, and Louis Steimeister

The major revisions have been made in red text.

Abstract: Microgrid is a promising technology of distributed energy supply system, which consists of storage devices, generation capacities including renewable sources, and controllable loads. It has been widely investigated and applied for residential and commercial end-use customers as well as critical facilities. In this paper, we propose a joint state-based dynamic control model on microgrids and manufacturing systems where optimal controls for both sides are implemented to coordinate the energy demand and supply so that the overall production cost can be minimized considering the constraint of production target. Markov Decision Process (MDP) is used to formulate the decision-making procedure. The main computing challenge to solve the formulated MDP lies in the co-existence of both discrete and continuous parts of the high-dimensional state/action space that are intertwined with constraints. A novel reinforcement learning algorithm that leverages both Temporal Difference (TD) and Deterministic Policy Gradient (DPG) algorithms is proposed to address the computation challenge. Experiments for a manufacturing system with an onsite microgrid system with renewable sources have been implemented to justify the effectiveness of the proposed method.

Key Words: Microgrid, Manufacturing, Reinforcement Learning, Markov Decision Process, Temporal Difference Learning, Deterministic Policy Gradient

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

Various studies on microgrids have been conducted for residential houses and critical facilities, such as medical centers, financial corporations, military bases, and jails (see details in Section 2.1). A great deal of literature focusing on the optimal control strategies of using microgrids has been reported (see details in Sections 2.2 and 2.3).

While for the manufacturing industry, the loss due to blackouts is not that direct or explicit when compared to the industries or sectors that are traditionally considered “critical”. The incurred losses such as unemployment and supply chain failure cannot be sensed and evaluated immediately after blackouts. The overall influence in term of both economic and societal aspects seems to be underestimated. Thus, the research on the application of microgrid in manufacturing has been less reported compared to residential sector.

The benefits of combining microgrids with manufacturing systems can be discussed from two aspects. On one hand, the microgrids can offer one more option of energy supply in terms of cost effectiveness. The cost of energy delivered by utility company varies depending on different time periods. The use of microgrids can provide a more flexible solution of energy supply considering the cost variation. The users can select different energy sources as well as determine energy flow to optimize their performance with respect to cost effectiveness. In addition, the use of microgrids can enhance the security of manufacturing systems against the utility failures
due to various reasons (e.g., natural disaster). On the other hand, microgrids can help manufacturing practitioners further improve their performance from the perspective of environmental sustainability. The adoption of renewable sources in microgrids can significantly improve the carbon footprints of the energy supplied for production.

Therefore, this study aims to enhance the use of microgrid in manufacturing sector through building a joint control model that can simultaneously adjust the energy supply and demand from both microgrid and manufacturing sides towards cost effectiveness and environment sustainability.

Recently, with the increasing concerns on climate change and environmental protection, the benefits of integrating energy-aware strategies in manufacturing have been recognized. Many sustainability-aware and carbon-constrained operating strategies, either on individual manufacturing processes or entire manufacturing system, have been investigated and used by many practitioners (see details in Section 2.4). Meanwhile, the use of microgrid to support manufacturing has also been launched, in particular, in the areas of benefits evaluation, system sizing, and microgrid-side control (see details in Sections 2.5 and 2.6), among which many investigations on using integrated energy systems for process industry have been reported (see details in Section 2.7).

However, the study of joint energy control for both microgrids and manufacturing systems simultaneously has not yet been fully launched. In this paper, a joint control model considering both microgrids and manufacturing systems is established using Markov Decision Process (MDP). A novel reinforcement learning algorithm is proposed for solving the MDP. The algorithm uses both Temporal Difference (TD) method to search for the discrete optimal control actions and Deterministic Policy Gradient (DPG) algorithms to search for the continuous optimal control actions, together with function approximations of the action-value function via a neural network. 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 towards cost optimality. It is empirically validated that the optimal policies found by the proposed reinforcement learning algorithm are more efficient in production and incur less cost when compared to randomly sampled policies and a routine operation policy. Last but not least, due to its exploration-exploitation nature, the proposed reinforcement learning method is also effective in the case when the MDP parameters are not known but have to be learned during the algorithm dynamics.

The proposed MDP model as well as the reinforcement learning algorithms can well address the major challenges with respect to modeling and problem solving. 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 as well as the external utility connections 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 also be cared when energy control for the manufacturing system is implemented. All these factors need to be carefully considered.

On the other hand, it can be expected that the space of the states and the actions in the MDP model could be extremely large, which makes most existing strategies for solving MDP such as reinforcement learning algorithms 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 [6].

In summary, the major contributions of the proposed joint control model are summarized as follows:

- In most relevant literature, the control is only focused on the microgrid side, while neglecting the demand side (i.e., manufacturing system). In some research where manufacturing side is integrated into the control scheme, the manufacturing system is usually simplified and thus the complex internal dynamics cannot be fully represented (see details in Section 2.6). In this paper, the dynamic joint controls of manufacturing systems and microgrids, e.g., the complex dynamics in the demand side when energy control is implemented on manufacturing systems, are modeled.
In existing literature, the renewable sources are usually modeled as the system state in MDP, that means, the renewable energy is dependent on the variations of wind and solar sources. It ignores the opportunities of controlling the energy supply from the renewable sources. In this paper, the availability of renewable sources is modeled as the state, while the ON/OFF control of renewable generation capacity is modeled as the control actions.

There exist a few published works that use reinforcement learning algorithms such as deep Q-learning (DQN) to solve dynamic decision-making problems with respect to microgrid operations [7, 8]. However, Q-learning algorithms is typically constrained by problem size, and thus cannot work very well if the problem size is too large. Our novel reinforcement learning algorithm integrates DPG with TD 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.

The remaining part of the paper is organized as follows. A brief literature review in several relevant areas is given in Section 2. Section 3 introduces the formulation of the dynamic decision-making model using MDP in detail. Section 4 proposes our reinforcement learning algorithm. Section 5 implements numerical case studies through extensive experiments and sensitivity analysis to validate the effectiveness of the proposed method. Section 6 concludes the paper and discusses future works. Supplementary materials in terms of mathematical derivations are summarized in Section 7.

2. Literature Review

2.1 Microgrids for Residential Properties, Community Services, and Critical Facilities

The research of microgrid applications in residential properties, community services, and critical facilities has been widely reported in the literature. For example, Faruque discussed an economically profitable way to deploy a residential microgrid incorporating a new market entity that can act as both a consumer and a power supplier [9]. Kriett and Salani proposed a generic mixed integer linear programming model to minimize the operating cost of a residential microgrid [10]. Roggia et al designed a sustainable residential microgrid system including PHEV and energy storage devices [11]. Ahourai and Faruque analyzed the impact of a residential microgrid under various electric vehicle penetration levels [12]. Igualada et al. proposed an optimization model to manage a residential microgrid including a charging spot with a vehicle-to-grid system and renewable energy sources [13]. Hawkes and Leach investigated the cost-effective operating strategy for residential micro-combined heat and power (CHP) systems [14]. Tasdighi et al. investigated a micro-CHP based residential microgrid scheduling problem using smart meter data and temperature dependent thermal load modeling [15]. Kakigano et al. applied a direct-current microgrid to residential houses with a cogeneration system, such as a gas engine or fuel cell [16].

2.2 Microgrid Control Strategies Using Traditional Methodologies

Considering the complex dynamics, in particular, the uncertainties of the renewable sources and the load, involved in microgrid operation, Model Predictive Control (MPC)-based approaches have been widely used to estimate the uncertainty and offer an optimizer to solve the best schedules of microgrid operations [17]. For example, an online optimal energy management model for energy storage system in a microgrid was developed using a mixed integer linear programming (MILP) over a rolling horizon period [18]. A similar method considering time-varying constraints of a microgrid was proposed [19]. MPC was applied to the operation of a hydrogen-based hybrid energy storage system in a microgrid [20]. A robust optimization model was proposed for microgrid operation using ensemble weather forecasts [21].

Another major research approach to address the challenges of uncertainties is stochastic optimization. For example, a stochastic energy scheduling model in microgrids with intermittent renewable energy resources was proposed [22]. A risk-averse stochastic programming method was proposed, which considered not only the expectation but the variation of the total cost [23]. A two-stage stochastic programming approach with MPC for microgrid energy management was proposed considering the uncertainty of load demand, renewable energy generation, and electricity prices [24]. From the perspective of control objectives, the existing literature focuses on optimizing cost [25, 26], power system stability [27, 28] such as voltage frequency control [29, 30], and environmental objectives such as reducing carbon dioxide emissions [31, 32].
2.3 Microgrid Control Strategies Using Machine Learning Methodologies

Recently, some researchers have started to leverage the methodologies such as deep learning and reinforcement learning to address the microgrid control problem, i.e., reinforcement learning framework is used to solve the sequential decision-making problem typically formulated by MDP, and deep learning is used to approximate the values of some critical components in MDP such as Q-value and the state value. For example, Zeng et al. presented a novel dynamic energy management tool that incorporates efficient management of energy storage system into microgrid real-time dispatch while considering power flow constraints and uncertainties in load, renewable generation, and real-time electricity price [7]. Ji et al. proposed a deep reinforcement learning approach to identify the optimal control strategy of microgrids that can minimize the daily operating cost. In their study, a deep feed-forward neural network is designed to approximate the optimal action-value function, and the deep Q-network algorithm is used to train the neural network [8].

2.4 Sustainability-Aware and Carbon-Constrained Manufacturing Control Strategy

Manufacturing activities dominate energy consumption and Greenhouse Gas (GHG) emissions in the industrial sector [33] that accounts for approximately one third of the total energy consumption in the U.S. [34]. Thus, many studies focusing on building energy consumption framework and exploring emission reductions for various industries have been reported. For example, a big data driven analytical framework was proposed to reduce the energy consumption and emission for energy-intensive industries [35, 36]. Therblig power model was developed for calculating the energy supply of Computer Numerical Control (CNC) machine tools using machining process parameters [37].

In addition, many sustainability-aware and carbon-constrained operating and control strategies have been investigated for manufacturing. Optimal energy control towards a smart and sustainable manufacturing paradigm has been widely reported. For example, a CPS-enabled and knowledge-aided demand response strategy was proposed for sustainable manufacturing [38]. Later, the same team investigated a demand response strategy for manufacturing systems considering the implications of fast-charging battery powered material handling equipment [39]. Work-in-process parts in manufacturing systems were utilized for optimal production control to reduce energy cost without sacrificing production throughput [40]. A joint energy control and maintenance scheduling model was proposed to minimize overall operational costs considering time-dependent energy cost as well as equipment degradation [41].

2.5 Microgrid System Sizing for Manufacturing

The benefits of using microgrids in manufacturing have been gradually recognized by both academic colleagues and industrial practitioners. 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 [42, 43]. Many studies focusing on the optimal design and component sizing of the microgrid for manufacturing plant has been reported [44-46]. 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 the manufacturing system and HVAC (Heating, Ventilation, and Air Conditioning) system in a typical manufacturing plant [44]. A time series model was proposed to describe and predict the variation of the energy load of manufacturing systems and the irradiation of solar energy such that a case study examining the cost for building and running an onsite microgrid system considering different microgrids capacities was implemented [46]. The onsite wind and solar power capacity that could maximize the cost saving of a manufacturing facility was estimated in an interruptible load demand response program [47].

2.6 Microgrid Control for Manufacturing

Some studies on using microgrid for manufacturing have been recently reported. For example, Harper et al. proposed a demand-side management method for manufacturing that uses a microgrid to reduce energy consumption and improve system reliability [48, 49]. They employed a simple rule-based control strategy to adjust the operation modes of industrial pumps based on the empirical parameters of supply capability and electricity demand. Golari et al. presented a multi-period, production-inventory planning model in a multi-plant
manufacturing system powered with onsite and grid renewable energy to determine the production quantity, the stock level, and the renewable energy supply in each period such that the aggregated production cost (including energy) is minimized [50]. It was formulated as a static optimization problem, rather than a dynamic state-based decision-making model.

The common limitations of the existing studies on microgrid control for manufacturing can be summarized as follows. 1) Manufacturing system is usually simplified and modeled as a single isolated machine, which ignores the complex dynamics and interactions of various equipment within the manufacturing system. 2) The manufacturing side is typically considered a fixed load, while neglecting its control flexibility. 3) The control strategies used for the manufacturing equipment are usually simple heuristic and/or empirical rule based utilizing the empirical parameters of supply capability and electricity demand.

2.7 Integrated Energy Systems in Process Industry

Process industries, e.g., food, pulp and paper, basic chemicals, refining, and iron/steel, are typically considered to be energy intensive. For example, the electricity demand of pulp & paper industry in 2020 was estimated to be 2.08EJ [51]. Many studies in terms of improving energy efficiency and reducing GHG emissions have been reported. For example, Pandey and Prakash systematically analyzed energy conservation opportunities in pulp & paper industry [52]. Ashok proposed a peak load management model for steel plant [53]. Later, he and his colleague generalized the previous reseach to an optimization model for industrial load management [54]. Ma et al. presented an architecture of energy-cyber-physical system enabled management for energy-intensive process industries to enhance the implementation of integrated energy system for a cleaner production strategy [55]. Later, this team proposed a big data-driven analytical framework for energy-intensive process industries towards sustainability [56, 57]. Zhang et al. explored the physical and chemical characteristics of fugitive emission sources in the iron and steel production process [58]. Sun et al. systematically analyzed the integrated optimization of material and energy flows in the iron and steel industry [59]. Zhang et al. proposed a carbon flow tracing and carbon accounting method for exploring CO\textsubscript{2} emissions of the iron and steel industry [60].

3. Markov Decision Process (MDP) for Joint Control of Manufacturing and Onsite Microgrid Systems

3.1 Formulation of Joint Energy Control Using MDP

3.1.1 Modeling Microgrids and Manufacturing Systems

An MDP model is proposed to model the decision-making of the joint control of both onsite microgrid systems and manufacturing systems. MDP has been widely used in modeling complex and evolving state-based decision-making problems [61, 62]. In this paper, 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. Note that, traditionally, microgrid meant a distributed energy generation system and renewable sources were not the default components. While, in recent years, with the increasing penetration and popularity of renewable sources in energy systems, the integration of wind and solar energy in microgrid has become the main stream in literature when studying microgrids, especially, considering the benefits with respect to environmental sustainability. Thus, renewable components are considered the default components in the microgrids modeled in this paper.

The manufacturing system modeled is a typical serial production line with \( N \) machines and \( N - 1 \) buffers as shown in Figure 2 (here \( N = 5 \), and work-in-progress parts are stored in buffers). Let \( i = 1, \ldots, N \) be the indexes of the machines and \( i = 1, \ldots, N - 1 \) be the indexes of the buffers.
Note that the type of the manufacturing system modeled in the paper is a flow shop setting. Many industries use such a layout for their production system, e.g., auto assembly system, aeroplane assembly system, etc. In practice, the system may be more complex, say, with some parallel configurations, while, a serial line is considered a fundamental layout from which many complex layouts can be studied through appropriate extensions.

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.

### 3.1.2 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 ($S^{mf}_{t}$), microgrid system ($S^{mic}_{t}$), and exogenous environmental features ($S^{env}_{t}$), which can be formulated by $S_{t} = (S^{mf}_{t}, S^{mic}_{t}, S^{env}_{t})$. $S^{mf}_{t}$ can be denoted by $S^{mf}_{t} = (S^{M}_{t1}, S^{M}_{tN}, S^{B}_{t1}, \ldots, S^{B}_{t(N-1)})$, where $S^{M}_{it}$ ($i = 1, \ldots, N$) denotes the state of machine $i$ in the manufacturing system at decision epoch $t$; $S^{B}_{it}$ ($i = 1, \ldots, N - 1$) 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 state, which is illustrated in Figure 3.
energy discharged by the battery for supporting manufacturing, and is given by a binary variable \( t = 1 \), not working\( = 0 \). A non-negative real number \( SOC_t \), denotes the state of charge of the battery system at decision epoch \( t \).

The state describing exogenous environmental features, \( S_t^{env} \), can be denoted by \( S_t^{env} = (I_t, \nu_t) \), where \( I_t \) denotes the solar irradiance at decision epoch \( t \), and \( \nu_t \) 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 \( (I_t, \nu_t) \) are taken from one year’s data (assumed to be 360 days and 24 hours/day, so in total 8640 hours).

### 3.1.3 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 \( (A_t^{mf}) \) and the microgrid system \( (A_t^{mic}) \), which can be denoted by \( A_t = (A_t^{mf}, A_t^{mic}) \). \( A_t^{mf} \) can be denoted by \( A_t^{mf} = (a_t^1, \ldots, a_t^N) \), where \( a_t^i \) \( (i = 1, \ldots, N) \) 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 \( Opr, Blo, Sta, Off \), and \( Brk \) 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 \( Opr, Blo, Sta \) states. \( W \)-action intends to turn on the machine that was previously turned off, which can only be applied to the machine in \( Off \) states. Note that for model simplicity, we assume that the energy consumption and time required for the transitions between different machine states can be ignored. The time and energy required for state transition depend on machine characteristics. For example, many CNC machines have a relatively short switch time as well as lower switch energy consumption when machine state is adjusted.

\( A_t^{mic} \) 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 \( A_t^{mic} = (a_t^g, a_t^w, a_t^b, s_t^m, s_t^b, w_t^m, w_t^b, g_t^m, g_t^b, p_t^m, p_t^b, h_t^m) \). Here \( a_t^g, a_t^w, \) and \( a_t^b \) 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; \( s_t^m, s_t^b, \) and \( s_t^{sb} \) 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 due to utility policies. Finally, \( h_t^m \) denotes the energy discharged by the battery for supporting manufacturing, and is given by a binary variable \( \delta_t^{bm} \), so that \( h_t^m = b \cdot \delta_t^{bm} \cdot \Delta t \) for some discharging rate \( b > 0 \). Note that the energy discharged by the battery is not considered for sold back.
3.1.4. State Transition
Let the function $P: S \times A \times S \rightarrow [0,1]$ be the transition probability function, so that $P(S', A, S) \equiv Pr(S'|S, A)$ is the probability of transition to state $S'$ given that the previous state was $S$ and action $A$ was taken. 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 (1) based on the states and the control actions adopted at decision epoch $t$ of upstream and downstream machines:

$$
S_{i(t+1)}^B = S_{it}^B + I(S_{it}^M, a_i^t) - I(S_{i+1}^M, a_{i+1}^t), 0 \leq S_{it}^B \leq N_i,
$$

(1)

Here $N_i$ is the capacity of buffer $i$, and $I(S_{it}^M, a_i^t)$ is an indicator function that is defined by (2):

$$
I(S_{it}^M, a_i^t) = \begin{cases} 
1 & \text{, when } S_{it}^M = Opr \text{ and } a_i^t = K \\
0 & \text{, when } S_{it}^M \neq Opr \text{ or } a_i^t = H
\end{cases}
$$

(2)

Referring to the literature focusing on the statistical methods for machine reliability [63], 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 (3) and (4) respectively:

$$
Pr(S_{i(t+1)}^M = Brk|S_{it}^M \neq Brk, S_{it}^M \neq Off) = Pr(L_i < t + \Delta t)
$$

(3)

$$
Pr(S_{i(t+1)}^M \neq Brk|S_{it}^M \neq Brk, S_{it}^M \neq Off) = Pr(L_i \geq t + \Delta t)
$$

(4)

Whether the machine is at Off state or not at next decision epoch can be determined in a deterministic way by (5):

$$
S_{i(t+1)}^M = Off \text{ if } (S_{it}^M = Off \text{ and } a_i^t = K) \text{ or } (S_{it}^M \neq Off \text{ and } a_i^t = H)
$$

(5)

In addition, we also assume $D_i$, which is the random repair time of machine $i$, follows Exponential distribution [64]. 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 (6) and (7) respectively.

$$
Pr(S_{i(t+1)}^M = Brk|S_{it}^M = Brk) = Pr(D_i < t + \Delta t)
$$

(6)

$$
Pr(S_{i(t+1)}^M \neq Brk|S_{it}^M = Brk) = Pr(D_i \geq t + \Delta t)
$$

(7)
From (3)-(7) we can decide whether the machine state $S_{i(t+1)}^M$ is \textit{Brk} or \textit{Off} or any one of the states \textit{Sta}, \textit{Blo}, \textit{Opr}. To further determine the exact state if we fall into one of the states \textit{Sta}, \textit{Blo}, \textit{Opr}, we use the following rules. For $2 \leq i \leq N$:

$$S_{i(t+1)}^M = \text{Sta} \text{ if } S_{i(t+1)}^M \neq \text{Brk/Off} \text{ and } S_{(i-1)(t+1)}^B = 0 \text{ and } S_{(i-1)(t+1)}^M = \text{Brk/Sta/Off}$$  

(8)

and for $1 \leq i \leq N - 1$:

$$S_{i(t+1)}^M = \text{Blo} \text{ if } S_{i(t+1)}^M \neq \text{Brk/Off} \text{ and } S_{i(t+1)}^B = N_i \text{ and } S_{(i+1)(t+1)}^M = \text{Brk/Blo/Off}$$  

(9)

and for all $1 \leq i \leq N$:

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

(10)

Notice that (8) and (9) can be used to find all the \textit{Sta} and \textit{Blo} state machines by forward/backward scanning of all machines starting from the first/last machine, due to the trivial fact that $S_{1(t+1)}^M \neq \text{Sta}$ and $S_{N(t+1)}^M \neq \text{Blo}$. With these, (10) can further help to determine \textit{Opr} state for machines.

Therefore, the system operation state transition between the current decision epoch and the next decision epoch can be calculated by using (1)-(10) when $A_{tmfg}$ is adopted based on a given $S_{tmfg}$.

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 (11) and (12), respectively:

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

(11)

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

(12)

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

The state transition of generator is determined by the control actions adopted, which can be formulated by:

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

(13)
The state transition of battery (i.e., \(SOC\)) is determined by the charging and discharging occurring between \(t\) and \(t+1\) as well as the original \(SOC\), which can be formulated by:

\[
SOC_{t+1} = SOC_t + (s_t^b + w_t^b + \eta - b_t^m) / \eta
\] (14)

where \(\eta\) is charging/discharging efficiency.

### 3.1.5. 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. At time \(t\), the cost at state \(S_t\) when action \(A_t\) is taken, can be defined by \(E(S_t, A_t)\). The total incurred cost from time 0 to the end of planning horizon, starting from state \(S\) and under policy \(\pi: S \rightarrow A\), is given by:

\[
C(\pi) = E[\sum_{t=0}^{\infty} \gamma^t E(S_t, \pi(S_t)) \mid S_0 = S]
\] (15)

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

For our model, the cost when action \(A\) is taken at state \(S\) is given by \(E(S, A)\), which is equal to energy consumption cost plus the microgrid operational cost, minus production throughput reward and the sold back reward. So, it can be calculated by:

\[
E(S, A) = TF(S, A) + MC(S, A) - TP(S, A) - SB(S, A)
\] (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.

Note that in this paper, since the throughput modeling and quantification for a typical manufacturing system is still a major research challenge in the field of production system engineering when considering non 100% reliable machines and finite capacity buffers [65, 66], the concern of production throughput maintaining 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.

In (16), \(TF(S, A)\) can be calculated by:

\[
TF(S, A) = p_t \cdot r_t^c = (p_t^m + p_t^b) \cdot r_t^c
\] (17)

where \(p_t\) is the energy consumption purchased from the grid at decision epoch \(t\), and \(r_t^c\) is the rate of energy consumption charge. \(p_t\) can be calculated by:
\[ p_t = E_t^{mf} - (s_t^m + w_t^m + g_t^m + b_t^m) \]  

(18)

where \( E_t^{mf} \) is the total energy consumed by the manufacturing system at decision epoch \( t \) which can be determined by:

\[ E_t^{mf} = \sum_{i=1}^{N} PC_{it} \cdot \Delta t \]  

(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:

\[
PC_{it} = \begin{cases} 
0 & \text{if } S_t^M = Brk \text{ or } S_t^M = Off \\
PC_{i}^{Opr} & \text{if } S_t^M = Opr \\
PC_{i}^{dl} & \text{if } S_t^M = Sta \text{ or } S_t^M = Blo 
\end{cases}
\]  

(20)

where \( PC_{i}^{Opr} \) and \( PC_{i}^{dl} \) are the power level of machine \( i \) at the states of \( Opr \) and \( Sta/Blo \), respectively.

\( MC(\mathbf{S}, \mathbf{A}) \) can be calculated by:

\[
MC(\mathbf{S}, \mathbf{A}) = e^s_t \cdot r^s_{omc} + e^w_t \cdot r^w_{omc} + e^g_t \cdot r^g_{omc} + \frac{(b^m_t + s^b_t + w^b_t + g^b_t) \cdot \Delta t}{2e(SOC_{max} - SOC_{min})} \cdot r^b_{omc},
\]  

(21)

where \( e^s_t, e^w_t, \) and \( e^g_t \) 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. \( r^s_{omc}, r^w_{omc}, \) and \( r^g_{omc} \) are the unit operational and maintenance cost for generating electricity from solar PV, wind turbine, and generator respectively. \( r^b_{omc} \) 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 allowed state of charge of battery storage system, respectively. Note that the fraction part of the 4th term on the right-hand side of (21) represents for the fraction of the charging/discharging cycle of the battery in one decision period [45]. Specifically, the numerator calculates the total energy flow (either charging or discharging) through the battery system in one decision period, while the denominator calculates the total energy flow of one charging/discharging cycle.

\( e^s_t \) can be calculated by:

\[
e^s_t = \begin{cases} 
0 & \text{if } g^s_t = 0 \\
I_t \cdot a \cdot \delta \cdot \Delta t/1000, & \text{if } g^s_t = 1
\end{cases}
\]  

(22)
where $I_t$ is the solar irradiance of a certain location (W/m$^2$) at decision epoch $t$, $a$ is the area of the solar PV system, and $\delta$ is the efficiency of the solar PV.

$e_t^w$ can be calculated by:

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

(23)

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 equipped within the onsite generation system and $R_{P_w}$ is the rated power of the wind turbine (kW).

$R_{P_w}$ is determined by:

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

(24)

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:

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

(25)

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:

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

(26)

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

$$
pr_t = \begin{cases} 
1, & \text{if } S^M_{Nt} = Opr \text{ and } a^N_t = K \\
0, & \text{if } S^M_{Nt} \neq Opr \text{ or } a^N_t = H
\end{cases}
$$

(27)
Note that, in such a serial production system as shown in Figure 2, the system throughput is counted by the production count of the last machine in the line since one finished product is only counted when the process of the last machine is completed. Thus, the subscript in (27) is $N$.

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

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

(28)

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

3.2 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 that the battery state of charge level needs to be maintained within a given range, which can be formulated by:

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

(29)

Notice that (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.,

$$a_t = \begin{cases} 
K, & \text{if } S_t^M = Opr, Blo, Sta, Off, Brk \\
H, & \text{if } S_t^M = Opr, Blo, Sta \\
W, & \text{if } S_t^M = Off
\end{cases}$$

(30)

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

$$s_t^m + s_t^b + s_t^{sb} = e_t^s,$$

(31)

$$w_t^m + w_t^b + w_t^{sb} = e_t^w$$

(32)

$$g_t^m + g_t^b + g_t^{sb} = e_t^g$$

(33)
Notice that according to (22), (23) and (25), the energies $e_1^x$, $e_1^w$, $e_1^g$ depend on the working states of the microgrid $(g_t^x, g_t^w, g_t^g)$, the solar irradiance $I_t$ and the wind speed $v_t$. The constraints (31)-(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:

$$ (s_t^b + w_t^b + g_t^b + p_t^b) \cdot b_t^m = 0. $$

(34)

Notice that since $b_t^m = \delta_t^{bm} \cdot b \cdot \Delta t$, we only seek for binary choices of $\delta_t^{bm} = 0/1$ 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:

$$ (s_t^{sb} + w_t^{sb} + g_t^{sb}) (p_t^m + p_t^b) = 0. $$

(35)

If the constraint (35) is satisfied at $s_t^{sb} = w_t^{sb} = g_t^{sb} = 0$, so that we allow $p_t^m + p_t^b \neq 0$, then due to supply-demand balance principle, the energy purchased from the grid should be equal to the energy consumed from the grid, so we have

$$ p_t^m + p_t^b = p_t 1_{\{p_t > 0\}}, $$

(36)

where $p_t$ is given by (18).

To simplify the model, we further assume that the energy purchased from the grid can only be used either for supporting manufacturing or charging battery, but not simultaneously, i.e.,

$$ p_t^m \cdot p_t^b = 0. $$

(37)

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

To facilitate the design of policy-gradient related algorithms for training, we further parameterize the control actions $(s_t^m, s_t^b, s_t^{sb}, w_t^m, w_t^b, w_t^{sb}, g_t^m, g_t^b, g_t^{sb})$ by introducing proportionality parameters.

$$ \theta = (\lambda_t^m, \lambda_t^b, \lambda_t^w, \lambda_t^m, \lambda_t^g, \lambda_t^b) $$

(38)
such that the proportionality relation holds as

\[
\begin{align*}
S_t^m &= e_t^s \cdot \lambda_s^m, \quad S_t^b = e_t^s \cdot \lambda_s^b, \quad S_t^{sb} = e_t^s \cdot (1 - \lambda_s^m - \lambda_s^b) \\
W_t^m &= e_t^w \cdot \lambda_w^m, \quad W_t^b = e_t^w \cdot \lambda_w^b, \quad W_t^{sb} = e_t^w \cdot (1 - \lambda_w^m - \lambda_w^b) \\
g_t^m &= e_t^g \cdot \lambda_g^m, \quad g_t^b = e_t^g \cdot \lambda_g^b, \quad g_t^{sb} = e_t^g \cdot (1 - \lambda_g^m - \lambda_g^b)
\end{align*}
\] (39)

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

\[
\begin{align*}
\lambda_s^m &\geq 0, \lambda_s^b \geq 0, 0 \leq \lambda_s^m + \lambda_s^b \leq 1 \\
\lambda_w^m &\geq 0, \lambda_w^b \geq 0, 0 \leq \lambda_w^m + \lambda_w^b \leq 1 \\
\lambda_g^m &\geq 0, \lambda_g^b \geq 0, 0 \leq \lambda_g^m + \lambda_g^b \leq 1
\end{align*}
\] (40-42)

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

\[
(\delta_t^b, \delta_t^p, \delta_t^{pb}, \delta_t^{sb}, \delta_t^{bm}) \in \{(1,1,1,0,0), (1,1,0,0,0), (1,0,0,1,0), (0,1,0,0,0), (0,1,0,0,1), (0,0,0,1,1)\}.
\] (43)

Notice that (43) summarizes all discrete constraints for the control parameters on the microgrid. The remaining continuous constraints for the microgrid are only (29) and (40)-(42). Based on (43), we can further write the constraints (29) and (40)-(42) into different constraints on \(\theta\) (the parameter in (38)) and \(SOC\). These constraints are formulated in (S1)-(S6) in Supplementary Material 7.3. All effective constraints for the admissible actions in this problem are (30), (43) and (S1)-(S6).


4.1 Abstract Formulation of the MDP Model

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

\[
S_t^d = (S_t^m, S_t^c, S_t^{fb}, S_t^{fb}, g_t^s, g_t^w, g_t^g, l_t, v_t)
\] (44)

which consists of the machine, buffer and microgrid states, as well as the coarse-grained solar irradiance and the wind speed \((l_t, v_t)\). Here to reduce complexity, an approximate coarse-grained scheme is applied to each pair of the values \((l_t, v_t)\), 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
\[ S_t^c = (SOC_t) \] (45)

which consists of the SOC state.

An action \( A_t \) in the action space \( A \) of the model also consists of three parts \( A_t = (A_t^d, A_t^c, A_t^r) \): the discrete part

\[ A_t^d = (a_t^1, \ldots, a_t^N, a_t^w, \delta_t^b, \delta_t^{pb}, \delta_t^{zb}, \delta_t^{bm}) \] (46)

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

\[ A_t^c = (s_t^m, s_t^{sb}, s_t^w, s_t^{wb}, s_t^{gb}, s_t^{gbm}) \] (47)

which consists of the solar/wind/generator energy used for supporting manufacturing, charging battery, and sold back to the grid. The continuous action \( A_t^c \) depends on \( S_t \) and the proportionality parameters in \( \theta \) introduced in (38), i.e., \( A_t^c = A^c(\theta_t, S_t) \); the remainder part \( A_t^r = (p_t^m, p_t^{pb}, b_t^m) \), 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 \( \delta_t^{pb} \) and \( \delta_t^{bm} \) as well as (36) and (18), which then can be calculated from \( A_t^c \).

At a specific state \( S_t \in S \), the actions that can be taken are restricted by this particular state via the restriction

\[ A_t^d \in D^d(S_t^d) \] (48)

And

\[ \theta \in D^c(S_t^d, S_t^c, A_t^d) \] (49)

where \( D^d \) is the set of admissible discrete actions \( A_t^d \) that can be taken at the current state, and \( D^c \) is the set of admissible parameters \( \theta \) for continuous actions \( A_t^c \) that can be taken at the current state. According to the discussions in Section 3.2, we see that \( D^d \) is given by (30) and (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 (S1)-(S6) in Supplementary Material 7.3, that depends on both the continuous and the discrete parts of the current state, as well as the discrete actions.

4.2 A Brief Overview of Reinforcement Learning: Temporal-Difference and Policy Gradient Methods

For an infinite horizon, discounted MDP with state space \( S \) and action space \( A \), Reinforcement Learning algorithms [67] design an “agent” interacting with an environment over a number of time steps. At each time
step $t$ when the agent is at a state $S_t$, it selects an action $A_t$ using the policy $\pi(S_t)$, and then moves to the next state $S_{t+1}$ according to the transition probability $P(S_{t+1}, A_t, S_t) = Pr(S_{t+1}|S_t, A_t)$. During this step the agent incurs a cost $E(S_t, A_t)$. The goal of the agent is to find a policy $\pi^*$ that minimizes the expected total discounted cost

$$\pi^* = \arg\min_{\pi} C(\pi)$$

(50)

where $C(\pi) \equiv E[\sum_{t=0}^{\infty} \gamma^t E(S_t, A_t)|S_0 = S]$, and $\gamma \in (0, 1]$ is the discount factor. To solve (50), we introduce the action-value function $Q^\pi(S, A)$, which is the expected total discounted cost under policy $\pi$ starting from state $S$ and taking initial action $A$. For the optimal policy $\pi^*$ in (50) we have $Q^\pi(S, A) \equiv Q^{\pi^*}(S, A) = \min_{\pi} Q^\pi(S, A)$. The key idea in action-value methods is to identify $Q^\pi(S, A)$ using a function approximator, such as a neural network $Q(S, A; \omega)$ built via an iterative scheme that approaches $Q^{\pi^*}(S, A)$. In practice we use a function approximator, such as a neural network $Q(S, A; \omega)$ (sometimes referred as the “critic”, see [67-70]) to replace the actual action-value function $Q^\pi(S, A)$, where $\omega$ is the parameter of the approximating function. Given current $\omega_t$, we construct the temporal difference

$$\delta_t = E(S_t, A_t) + \gamma Q(S_{t+1}, A_{t+1}; \omega_t) - Q(S_t, A_t; \omega_t)$$

(51)

where $S_{t+1}$ is the next state of MDP from state $S_t$ under action $A_t$ and $A_{t+1}$ is selected as the optimal policy (or near-optimal policy), such as $\varepsilon$-greedy policy [67] at state $S_{t+1}$ when the approximator is given by the parameter $\omega_t$. This class of action-value methods are called (on-policy) Temporal Differences (TD or SARSA, see [67]). Actually, there are two kinds of reinforcement learning methods: on-policy and off-policy. On-policy methods attempt to evaluate or improve the policy that is used to make decisions, whereas off-policy methods evaluate or improve a policy different from that used to generate the data (see [67, Section 5.4, p. 82]). In our temporal-difference method, we are using the policy obtained from data in the current iteration step to generate new data and we repeat this iteration. So, our TD method is on-policy. Once the temporal differences (51) are constructed, we can update the critic parameter $\omega_t$ according to a standard gradient descent

$$\omega_{t+1} = \omega_t - \eta_\omega \delta_t V_\omega Q(S_t, A_t; \omega_t) ,$$

(52)

where $\eta_\omega > 0$ is the learning rate. Another way to solve (50) works in the scenario when the policy can be parameterized by a policy parameter $\theta$, so that $\pi = \pi_\theta$, $C(\pi) = C(\pi_\theta) = C(\theta)$. In that case, we can simply use a gradient descent to find the optimal policy parameter $\theta^*$ such that $\pi^* = \pi_{\theta^*}$:

$$\theta_{t+1} = \theta_t - \eta_\theta V_\theta C(\theta)$$

(53)

where $\eta_\theta > 0$ is the learning rate. The gradient $V_\theta C(\theta)$ can be estimated by the well-celebrated policy gradient theorem when the policy $\pi_\theta$ is random (see [67]). However, if we are restricted only to deterministic policies, so that we assume that $\pi_\theta(S) = A(\theta)$ is a deterministic policy, and thus $C(\theta) = C(A(\theta))$, then the Deterministic Policy Gradient (DPG, see [68]) can also be calculated as:
\[ \mathbb{V}_\theta C(\theta) = E_{S \sim \rho_\pi} \left[ \mathbb{V}_\theta A(\theta) \mathbb{V}_A Q^\pi(S, A) | \pi = A(\theta), A = A(\theta) \right] \]  

(54)

where \( \rho_\pi |_{\pi = A(\theta)} = E_{S_0} \sum_{t=1}^{\infty} \gamma^t P_r(S_t = S | S_0, \pi) \) is the discounted state distribution when \( \pi = A(\theta) \).

4.3 Neural-Network Integrated Reinforcement Learning for Solving the Optimal Policy of the Model in its Abstract Formulation

Based on the abstract formulation in Section 4.1, a policy \( \pi \) in the objective function (15) can be written as \( \pi(S_t) = \left( A^d_t(S_t), A^c(\theta_t, S_t), A^r_t(S_t) \right) \). The purpose is then to find optimal proportionality parameter \( \theta^* \) and for each state \( S \in S \), to identify the optimal control actions \( \left( A^d(S), A^c(\theta, S), A^r(S) \right) \), so that the objective function (15) can be minimized. Notice that, if \( \theta \) is the only parameter for the control actions, then DPG can be used to search for an optimal \( \theta \). On the other hand, if we only have to search for the optimal discrete actions \( A^d(S) \), then we can employ TD. Since we are simultaneously targeting at these two tasks, we alternatively perform these two optimization steps. We take into account the constraints \( D(S) \), so that at the TD step, we search over the constraint \( D^c(S) \) on the discrete action space \( A^d \), while at the policy gradient step, we project the gradient updates to the continuous constraint \( D^c(S) \) on \( A^c \) using proximal projection operators (see [72]). 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.

The proposed method is model-free (see [67]), as compared with model-based methods (see [73]). This means that we do not have to know in advance any of the MDP system parameters. Rather, the exploration mechanism in the reinforcement learning algorithm will help us to estimate all the system parameters in an implicit way and make use of them. In the case study in Section 5, the model parameters are given only in order to simulate the system, but the learning algorithm does not require to know these parameters.

Below we write the proposed algorithm into pseudo-code at Algorithm 1.

**Algorithm 1** Reinforcement Learning via integrating TD control and DPG with Proximal Projection

1. **Input**: Input state space \( S \), action space \( A \), constraints \( D^d, D^c \); Given the neural network architecture \( Q(S, A^d, A^c(\theta), A^r; \omega) \); Discount factor \( 0 < \gamma < 1 \); Learning rates \( \eta_\theta, \eta_\omega > 0 \);

2. **Initialization**: Initialize the weight vector \( \omega_0 \sim \) a given prior distribution; initial action \( A^d_0, A^c_0 \) and action parameter \( \theta_0, A^r_0 = A^c(\theta_0) \); Initial state \( S_0 = (S^d_0, S^c_0) \);

3. for \( t = 1, 2, ..., T \) do

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

5. Calculate the total cost \( E(S_t, A_t) \);

6. Identify

\[ A^d_{t+1} = \arg \min_{A^d \in D^d(S_{t+1})} Q(S_{t+1}, A^d, A^c(\theta_t), A^r_t; \omega_t) ; \]

7. Based on \( A^d_{t+1} \), update the policy parameter \( \theta \) according to DPG (53) and (54):

\[ \theta_{t+1} = \text{prox}_{\gamma(S_{t+1}, A^d_{t+1})} [\theta_t - \eta_\theta \nabla_\theta Q^c(\theta_t) \nabla A^c(Q(S_t, A^d_t, A^c_t, A^r_t; \omega_t)) ; \]

8. Based on \( A^d_{t+1} \) and \( \theta_{t+1} \), obtain \( A^r_{t+1} \), so that

\[ A_{t+1} = (A^d_{t+1}, A^c_{t+1} = A^c(\theta_{t+1}), A^r_{t+1}) ; \]
9: Calculate TD (51):
\[
\delta_t = E(S_t, A_t) + \gamma Q(S_{t+1}, A_{t+1}; \omega_t) - Q(S_t, A_t; \omega_t);
\]

10: Update the weight vector \( \omega \) of the critic based on (52):
\[
\omega_{t+1} = \omega_t - \eta_\omega \delta_t \nabla Q(S_t, A_t; \omega_t);
\]

11: end for

12: Output: With the given optimal \( \omega^* \) and \( \theta^* \) found, for each state \( S \) given, output the approximate optimal policy \( (A^d, A^c(\theta^*), A^r) \), where
\[
(A^d, A^r) = \arg \min_{A^d, A^r \text{admissible}} Q(S, A^d, A^c(\theta^*), A^r; \omega^*).
\]

The flow chart of Algorithm 1 is shown in Figure 4 below.

**Figure 4:** The flow chart of Algorithm 1

### 4.4 Practical Implementation Details of Algorithm 1 at Original MDP Model

When implementing Algorithm 1 at the original MDP model for the microgrid-manufacturing system, practical issues need to be addressed. The set of admissible parameters \( \theta \in D^c(S, A) \) for continuous actions \( A^c \), that are determined by (S1)-(S6) in Supplementary Material 7.3, have an intersection structure. Indeed from (S1)-(S6) one can view \( D^c(S, A) \) as an intersection \( D^c(S, A) = D^c \cap D^{SOC} \left( \theta, SOC_t, (\delta_t^b, \delta_t^p, \delta_t^{pb}, \delta_t^{sb}, \delta_t^{bm}) \right) \). Here we set the fixed simplex

\[
D^c = \{(\lambda_s^m, \lambda_s^b, \lambda_w^m, \lambda_w^b, \lambda_g^m, \lambda_g^b); \lambda_s^m, \lambda_s^b, \lambda_w^m, \lambda_w^b, \lambda_g^m, \lambda_g^b \geq 0, 0 \leq \lambda_s^m + \lambda_s^b \leq 1, 0 \leq \lambda_w^m + \lambda_w^b \leq 1, 0 \leq \lambda_g^m + \lambda_g^b \leq 1\}
\]

(55)

The complicacy in \( D^c(S, A) \) lies in the other part of the intersection \( D^{SOC} \left( \theta, SOC_t, (\delta_t^b, \delta_t^p, \delta_t^{pb}, \delta_t^{sb}, \delta_t^{bm}) \right) \), that may vary according to the choice of \( (\delta_t^b, \delta_t^p, \delta_t^{pb}, \delta_t^{sb}, \delta_t^{bm}) \) 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 (S1)-(S6). But we can then fix this issue by using the $\theta$ found on the relaxed constraint set $D^c$ and determine the binary variables $(\delta_t^b, \delta_t^p, \delta_t^{pp}, \delta_t^{sb}, \delta_t^{bm})$, as well as update the SOC. If the SOC values we obtain violate the additional constraints in $D^{SOC}(\theta, SOC_t_(\delta_t^b, \delta_t^p, \delta_t^{pp}, \delta_t^{sb}, \delta_t^{bm}))$, we will just set them to be the boundary values $SOC_{max}$ or $SOC_{min}$, so that they will not violate the SOC constraints. In this way, we can find an approximate solution to the optimal one, with computationally achievable simulations.

Under this simplification, the solution algorithm for the original model becomes practical to implement. We can then change Algorithm 1 accordingly, so that

- In Step 6 of Algorithm 1, we select one action $A_{t+1}^d$ that only takes into account the constraint (30) and minimizes the $Q$-value. The admissible actions $A^d$ are stored in a tree constructed based on (30) and we search over the tree to identify optimal action $A_{t+1}^d$ based on Step 6. The action $A_{t+1}^d$ does not include the indicator variables $(\delta_{t+1}^b, \delta_{t+1}^p, \delta_{t+1}^{pp}, \delta_{t+1}^{sb}, \delta_{t+1}^{bm})$ in (43).
- In Step 7 of Algorithm 1, the constraint $D^c(S_{t+1}, A_{t+1}^d)$ is replaced by a fixed constraint $D^c$ in (55). Notice that in this case, the projection onto $D^c$ can be calculated directly, see Supplementary Material 7.1.
- Once $\theta$ is chosen, we determine whether or not we should choose $p_t^{m} + p_t^{b} \neq 0$, and whether $p_t^{b} \neq 0$ according to (35), (36), and (37); we also determine whether or not we choose $b_t^{m} \neq 0$ according to (34). The precise scheme of choosing $(p_t^{m}, p_t^{b}, b_t^{m})$ can be found in Supplementary Material 7.2.
- With all these ready, Step 8 in Algorithm 1 will be modified accordingly.
- At each loop, the optimal actions are implemented for the MDP system to jump to the next state and the incurred cost, throughput and energy demand are calculated.

5. Case Study
5.1 Experimental Parameters
5.1.1 Joint Microgrid-Manufacturing System Parameters
In this section, numerical case studies are implemented to illustrate the benefits of the proposed modeling and solution strategies. The open-source code for the proposed model in this paper is released at the GitHub repository [74].

We have carried out experiments of the microgrid-manufacturing system using a real-case parameter set. The manufacturing system includes five machines and four buffers as shown in Figure 2. We summarize all the system parameters in Tables 1-5. The parameters of the manufacturing system are taken according to [78], where machine and buffer parameters of the manufacturing system are shown in Table 1 and Table 2, respectively. Note that the mean time between failures of each machine is modeled as random variables following 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 $10^5$ per unit.

Table 1. Machine Parameters

<table>
<thead>
<tr>
<th>Machine</th>
<th>Mean time between failures (min)</th>
<th>Scale/Shape Parameter</th>
<th>Mean time to repair (min)</th>
<th>Rated power of operation (kW)</th>
<th>Power at idle state (kW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>111.39 min/1.5766</td>
<td></td>
<td>4.95</td>
<td>115.5</td>
<td>105</td>
</tr>
<tr>
<td>M2</td>
<td>51.1 min/1.6532</td>
<td></td>
<td>11.7</td>
<td>115.5</td>
<td>105</td>
</tr>
<tr>
<td>M3</td>
<td>110.9 min/1.7174</td>
<td></td>
<td>15.97</td>
<td>115.5</td>
<td>105</td>
</tr>
<tr>
<td>M4</td>
<td>239.1 min/1.421</td>
<td></td>
<td>27.28</td>
<td>170.5</td>
<td>155</td>
</tr>
<tr>
<td>M5</td>
<td>122.1 min/1.591</td>
<td></td>
<td>18.37</td>
<td>132</td>
<td>120</td>
</tr>
</tbody>
</table>
Table 2. Buffer Parameters

<table>
<thead>
<tr>
<th></th>
<th>B₁</th>
<th>B₂</th>
<th>B₃</th>
<th>B₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capacity</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Initial</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3. Wind Turbine Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v^{ci}$ (m/s)-Cut in speed</td>
<td>3</td>
<td>$\eta_l$-Gearbox efficiency</td>
<td>0.9</td>
</tr>
<tr>
<td>$v^{co}$ (m/s)-Cut off speed</td>
<td>11</td>
<td>$\eta_g$-Generator efficiency</td>
<td>0.9</td>
</tr>
<tr>
<td>$v^r$ (m/s)-Rate speed</td>
<td>7</td>
<td>$\theta$-Power coefficient</td>
<td>0.593</td>
</tr>
<tr>
<td>$\rho$ (kg/m$^3$)-Air density</td>
<td>1.225</td>
<td>$r_{omc}^w$ ($/kWh$)</td>
<td>0.08</td>
</tr>
<tr>
<td>$r$ (m)-Blade radius</td>
<td>25</td>
<td>$N_w$ (unit)</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4. Battery Storage Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e$ (kWh)-Capacity</td>
<td>350</td>
<td>$b$-Charging rate (kW)</td>
<td>2</td>
</tr>
<tr>
<td>$SO_{C_{max}}$ (%)</td>
<td>95</td>
<td>$SO_{C_{min}}$ (%)</td>
<td>5</td>
</tr>
<tr>
<td>$r_{omc}^b$ ($/kWh$)</td>
<td>0.9</td>
<td>$\eta$-Efficiency</td>
<td>0.99</td>
</tr>
</tbody>
</table>

Table 5. Solar Panel and Generator Parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$ (m$^2$)-Panel area</td>
<td>1400</td>
<td>$\eta_g$ (unit)-Number of generator</td>
<td>1</td>
</tr>
<tr>
<td>$\delta$-Efficiency</td>
<td>0.2</td>
<td>$G_p$ (kW)-Generator capacity</td>
<td>650</td>
</tr>
<tr>
<td>$r_{omc}^s$ ($/kWh$)</td>
<td>0.17</td>
<td>$r_{omc}^d$ ($/kWh$)-Operating cost</td>
<td>0.45</td>
</tr>
</tbody>
</table>

The parameters of the microgrid used in the experiment are sized based on the manufacturing load according to the methods proposed in [45]. The parameters related to wind turbine, battery storage system, and solar panel and generator are illustrated in Table 4, Table 5, and Table 6, respectively. The data of solar irradiance and wind speed are collected from Solar Energy Local [76] and State Climatologist of Illinois [77], respectively.

In order to prevent computational overflow, in the 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 min = 3600 s), speed measured by km/h ($3.6$ m/s), energy measured by MegaWatt ($10^6$ Watt), money cost measured by $10^4$ USD, area measured by km$^2$ ($10^6$ m$^2$), mass measured by $10^6$ kg. Time period is measured in hours.

5.1.2 Reinforcement Algorithm Parameters

The neural network $Q(S, A; \omega)$ is taken to be fully connected and it 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 $\eta_\theta = 0.003$ and $\eta_\omega = 0.0003$, with $\eta_\omega$ discounted by a factor 0.999 at each iteration.

Figure 5-(a) plots the $L^2$-norm of the difference at two consecutive iterations during training in the neural network weights ($\| \omega_{t+1} - \omega_t \|_2^2$), which clearly indicates the convergence of the training process. Figure 5-(b) plots the cumulative reward function (total incurred cost $C(S, \pi)$ in (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) [6]. After discretization of the continuous states and actions, the state space has a size of $3.8 \times 10^3$ and the action space has a size of $2.6 \times 10^4$. We use a smaller fully-connected neural network $Q(S, A; \omega)$ 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., $\| \omega_{t+1} - \omega_t \|_2^2$). The discount factor $\gamma = 0.1$. The neural network is trained using Adam [75] with different learning rates: Figure 5-(c) is for learning rate 0.001 and Figure 5-(d) is for learning rate 0.0001. It is seen that in these two cases, even after $10^4$ iterations, the vanilla $Q$-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 $(\omega^*, \theta^*)$ found for the neural-network $Q(S, A; \omega^*)$ and the continuous action $A^*(\theta^*)$ (see Algorithm 1), we tested the corresponding microgrid-manufacturing model at a time horizon of 100 time periods, with each time period equals one hour.

### 5.1.3. Baseline Scenarios

The results are compared with two baseline scenarios: The first one runs under a random policy, while the second one is a routine policy. For the random policy, the accumulation of total incurred cost $E(S, A)$ at (16), total energy cost $TF(S, A)$ at (17) and total production units $pr$ in (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).

For the routine policy, we consider a routine 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:

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

where $T$ is the set including all time periods $t$, $p_i$ is the 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 as the rated power without considering the difference between \( PC_i^{opr} \) and \( PC_i^{tl} \).

Two constraints are formulated as follows:

\[
\sum_{t \in T} x_{Nt} \cdot PR_N \geq TA
\]  

(57)

where \( x_{Nt} \) 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.

\[
0 \leq B_i(t+1) = B_i + x_{it} \cdot PR_i - x_{i(t+1)} \cdot PR_{i+1} - C_i
\]  

(58)

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 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 following the 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 routine 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 higher priority to be used than solar source since the cost of wind energy is typically lower than solar energy.

To match the target production unit made by optimal policy, for this routine strategy found by Integer Programming, we set the target output at the time horizon 100, i.e., the target production unit \( TA \) in (56) to be equal to 73, which is the total production throughput in units for the optimal policy (the quantity \( pr_t \) in (26)) at time horizon 100.

5.2 Result Analysis

5.2.1 Random Strategy

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 6, 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 \( pr_t \) in (26) of 73, while the randomly selected policy only produces 24. At the same time, the total cost for the optimal policy is \(-$728,293\) and the total cost for the random policy is \(-$236,214\). In terms of energy cost, optimal policy is also about one time less than the random policy, with \$876 for optimal policy and \$1,656 for random policy. The two other experiments behave very similarly.

5.2.2 Routing Strategy

We found that the total energy cost under the optimal policy found by this integer programming is \$3,642. This has been about four times of our previously announced \$876 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 6, where red solid lines are for the optimal policy selected by reinforcement learning and blue dashed lines are for the routine strategy found by an integer programming. It is clearly seen that the reinforcement learning model selects a policy that incurs less energy cost.

5.2.3 Comparisons
The overall comparison between proposed reinforcement learning model and random policy as well as routine policy is summarized in Table 6, where the last column also compares the average training time (CPU time over 3 experiments) consumed in training for the routine strategy and our proposed reinforcement learning method in the paper.

<table>
<thead>
<tr>
<th></th>
<th>Energy Cost ($)</th>
<th>Production (unit)</th>
<th>Throughput</th>
<th>Training CPU Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Routine Strategy</td>
<td>3,642</td>
<td>73</td>
<td></td>
<td>29</td>
</tr>
<tr>
<td>Proposed Model</td>
<td>876</td>
<td>73</td>
<td></td>
<td>457</td>
</tr>
<tr>
<td>Random Policy</td>
<td>1,656</td>
<td>24</td>
<td></td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 6. Comparison Among Three Models

Figure 6. Left to Right: Average over three experiments the comparison of (a) total throughput in production unit; (b) total energy cost incurred by optimal, routine and random policies: red solid line = optimal policy, blue dashed line = routine strategy via integer programming, black star line = random policy.

6. Conclusion
This paper proposes a joint dynamic control model for microgrids and manufacturing systems using 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 TD and DPG algorithms is proposed to solve the joint control of microgrid and manufacturing system towards cost optimality. Experiments for a manufacturing system with an onsite microgrid with renewable sources have been implemented and the results show the effectiveness of the proposed method in addressing the “curse of dimensionality” in dynamic decision-making with high dimensional and complicated state and action spaces.

There are several remarks and future directions that we would like to address here: (1) One can consider real
time decision making for emergency situations such as natural disasters that lead to non-availability of external energy supplies from grids (i.e., some new relevant constraints need to be added); (2) Our method could also be extended to other types of manufacturing systems, e.g., flexible manufacturing system, for an extended application scope; (3) The time consumed for the training process in reinforcement learning can be reduced by using reinforcement learning schemes with more exploration, such as $\varepsilon$-greedy [67], Monte-Carlo tree search [79] or upper-confidence-bound method [80].

7. Supplementary Material

7.1 The Projection onto the Simplex $D^c$

In this section we calculate directly the projection operator $\text{prox}_{D^c}(\theta)$ for a given $\theta \in \mathbb{R}^5$ onto the simplex

$$D^c = D_x^c \times D_w^c \times D_g^c,$$

where $D_x^c, D_w^c, D_g^c$ 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

$$\text{prox}_{D^c}(\theta) = \left(\text{prox}_{D_x^c}(\theta_s), \text{prox}_{D_w^c}(\theta_w), \text{prox}_{D_g^c}(\theta_g)\right).$$

If $\theta_0 \in D$, then $\text{prox}_D(\theta_0) = \theta_0$. For each $\theta_0 = (\theta_0^1, \theta_0^2) \in \mathbb{R}^2 \setminus D$, we can easily calculate $\text{prox}_D(\theta_0)$ according to the following rules

- $\theta_0^1 < 0, \theta_0^2 < 0$, then $\text{prox}_D((\theta_0^1, \theta_0^2)) = (0,0)$;
- $0 \leq \theta_0^1 \leq 1, \theta_0^2 < 0$, then $\text{prox}_D((\theta_0^1, \theta_0^2)) = (\theta_0^1, 0)$;
- $\theta_0^1 > 1$ and $\theta_0^1 - \theta_0^2 > 1$, then $\text{prox}_D((\theta_0^1, \theta_0^2)) = (1,0)$;
- $\theta_0^1, \theta_0^2 > 0, \theta_0^1 + \theta_0^2 \geq 1$ and $-1 \leq \theta_0^1 - \theta_0^2 \leq 1$, then $\text{prox}_D((\theta_0^1, \theta_0^2)) = \left(\frac{1 + \theta_0^1 - \theta_0^2}{2}, \frac{1 - \theta_0^1 + \theta_0^2}{2}\right)$;
- $\theta_0^2 > 1$ and $\theta_0^1 - \theta_0^2 < -1$, then $\text{prox}_D((\theta_0^1, \theta_0^2)) = (0,1)$;
- $0 \leq \theta_0^2 \leq 1, \theta_0^1 < 0$, then $\text{prox}_D((\theta_0^1, \theta_0^2)) = (0, \theta_0^2)$.

7.2 The Scheme for Choosing Battery Charging/Discharging Actions

Once $\theta$ is chosen, we determine whether or not we should choose $p_{t+1}^m + p_{t+1}^b \neq 0$, and whether $p_{t+1}^b \neq 0$ according to (35), (36), and (37); we also determine whether or not we choose $b_{t+1}^m \neq 0$ according to (34). To be precise, we first calculate the vector:

$$T = \left(\mathbf{1}_{\{s_{t+1}^{cb} + w_{t+1}^{cb} + g_{t+1}^{cb} > 0\}}, \mathbf{1}_{\{s_{t+1}^{cb} + w_{t+1}^{cb} + g_{t+1}^{cb} > 0\}}, \mathbf{1}_{\{SOC_{t+1} - b\Delta t/\eta - SOC_{min} \geq 0\}}\right),$$

where $s_{t+1}^{cb}, w_{t+1}^{cb}$ and $g_{t+1}^{cb}$, $s_{t+1}^{wb}, w_{t+1}^{wb}$ and $g_{t+1}^{wb}$ can be calculated according to $\theta_{t+1}$ updated in Step 7 of Algorithm 1, and $SOC_{t+1}$ is from $S_{t+1}$. Then, we determine $p_{t+1}^m, p_{t+1}^b$ and $b_{t+1}^m$ according to Table S1.

<table>
<thead>
<tr>
<th>$T$</th>
<th>(1,1,1)</th>
<th>(1,1,0)</th>
<th>(1,0,1)</th>
<th>(1,0,0)</th>
<th>(0,1,1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{t+1}^m$</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$\hat{p}_{t+1}/0$</td>
</tr>
</tbody>
</table>
Based on (43), we can further write the constraints (29) and (40)-(42) into different constraints on \( \theta \) (the parameter in (38) and \( SOC \)). These constraints are listed as follows:

- \( (\delta^b_t, \delta^p_t, \delta^{pb}_t, \delta^{sb}_t, \delta^{bm}_t) = (0,0,0,1,1) \). The constraints on \( \theta \) are given by:
  \[
  \begin{align*}
  \lambda^b_s &= \lambda^b_w = \lambda^b_g = 0 \\
  0 &\leq \lambda^m_s, \lambda^m_w, \lambda^m_g < 1 \\
  SOC_{min} &\leq SOC_t - b\Delta t/\eta \leq SOC_{max}
  \end{align*}
  \]
  \hspace{0.2cm} (S1)

- \( (\delta^b_t, \delta^p_t, \delta^{pb}_t, \delta^{sb}_t, \delta^{bm}_t) = (0,1,0,1,1) \). The constraints on \( \theta \) are given by:
  \[
  \begin{align*}
  \lambda^b_s &= \lambda^b_w = \lambda^b_g = 0 \\
  \lambda^m_s &= \lambda^m_w = \lambda^m_g = 1 \\
  SOC_{min} &\leq SOC_t - b\Delta t/\eta \leq SOC_{max}
  \end{align*}
  \]
  \hspace{0.2cm} (S2)

- \( (\delta^b_t, \delta^p_t, \delta^{pb}_t, \delta^{sb}_t, \delta^{bm}_t) = (0,1,1,0,0) \). The constraints on \( \theta \) are given by:
  \[
  \begin{align*}
  \lambda^b_s &= \lambda^b_w = \lambda^b_g = 0 \\
  \lambda^m_s &= \lambda^m_w = \lambda^m_g = 1 \\
  SOC_{min} &\leq SOC_t + \eta \left( E_{mf_g} - (s^m_t + w^m_t + g^m_t + b\Delta t) \right) \\
  &\cdot 1 \left( E_{mf_g} - (s^m_t + w^m_t + g^m_t + b\Delta t) > 0 \right) \leq SOC_{max}
  \end{align*}
  \]
  \hspace{0.2cm} (S3)

- \( (\delta^b_t, \delta^p_t, \delta^{pb}_t, \delta^{sb}_t, \delta^{bm}_t) = (1,1,0,0,0) \). The constraints on \( \theta \) are given by:
For convenience of the reader, we have summarized all the notations and symbols used in this paper as follows:

7.4 Summary of Notations and Symbols

For convenience of the reader, we have summarized all the notations and symbols used in this paper as follows:

**Actions and Policy**

- **A**: Action space
- **π**: Policy mapping S → A
- **A**: An action A ∈ A
- **A_t**: Action at time t
- **A^mfg**: Action for the manufacturing system
- **A^mic**: Action for the microgrid system
- **a^i_t**: Control action for machine i ∈ {1, ..., N} at time t
- **K**: Keep original machine state
- **W**: Turn a machine, which was Off, on
- **H**: Turn a machine Off
- **a^s_t,a^w_t,a^g_t**: Action of adjusting working status of the solar, wind, and the generators in the microgrid
- **s^m_t,s^b_t,s^{rb}_t**: Solar energy used to support manufacturing, charging the battery, and selling back to the grid, respectively
- **w^m_t,w^b_t,w^{rb}_t**: Wind energy used to support manufacturing, charging the battery, and selling back to the grid, respectively
- **g^m_t,g^b_t,g^{rb}_t**: Generator energy used to support manufacturing, charging the battery, and selling back to the grid, respectively
- **p^m_t,p^b_t**: Energy purchased from the grid for manufacturing and charging the battery, respectively
- **b^m_t**: Energy discharged by the battery to support manufacturing
- **δ^m_t**: Binary variable indicating whether the battery is discharged to support manufacturing or not

### Constraints

- \( (\delta^b_t, \delta^p_t, \delta^{rb}_t, \delta^s_t, \delta_t^{bm}) = (1,0,0,1,0) \)
  - \( (s^m_t + w^b_t + g^b_t) \leq SOC_{max} \)

- \( (\delta^b_t, \delta^p_t, \delta^{rb}_t, \delta^s_t, \delta_t^{bm}) = (1,1,1,0,0) \)
  - \( (s^m_t + w^b_t + g^b_t) \leq SOC_{max} \)

### Definitions

\( SOC_{min} \leq SOC_t \)

\( (S4) \)

\[ \begin{align*}
\lambda^m_t &\geq 0, \lambda^b_t > 0, \lambda^m_s + \lambda^b_s = 1 \\
\lambda^m_w &\geq 0, \lambda^b_w > 0, \lambda^m_w + \lambda^b_w = 1 \\
\lambda^m_g &\geq 0, \lambda^b_g > 0, \lambda^m_g + \lambda^b_g = 1 \\
SOC_{min} &\leq SOC_t + \eta(s^b_t + w^b_t + g^b_t) \leq SOC_{max}
\end{align*} \]

\( (S5) \)

\[ \begin{align*}
\lambda^m_t &\geq 0, \lambda^b_t > 0, \lambda^m_s + \lambda^b_s = 1 \\
\lambda^m_w &\geq 0, \lambda^b_w > 0, \lambda^m_w + \lambda^b_w = 1 \\
\lambda^m_g &\geq 0, \lambda^b_g > 0, \lambda^m_g + \lambda^b_g = 1 \\
SOC_{min} &\leq SOC_t + \eta\big((s^b_t + w^b_t + g^b_t) \\
&+ \left(E_{mf} - (s^m_t + w^m_t + g^m_t) + b\Delta t\right) \cdot 1_{(E_{mf} - (s^m_t + w^m_t + g^m_t) + b\Delta t) > 0}\big) \leq SOC_{max}
\end{align*} \]

\( (S6) \)
Parameterization of Action Space & Constraints

Positive battery discharge rate $b$: $\lambda^m, \lambda^p, \lambda^w, \lambda^g, \lambda^q$: Proportionality parameters in $[0,1]$, sum in each index less than or equal to 1

$S^d_t$: Discrete states at time $t$

$S^c_t$: Continuous states at time $t$

$A^d_t$: Discrete actions at time $t$

$A^c_t$: Continuous actions at time $t$

$A^r_t$: Remaining actions at time $t$

$D^d(S^d_t)$: Set of admissible discrete actions

$D^c(S^c_t, A^d_t)$: Set of admissible continuous actions

$\delta_t$: On-policy temporal difference

$\theta$: Parameters for the continuous part of the policy

$x_{it}$: Decision variable for machine $i$ at time $t$

$C_i$: Capacity of the $i$-th buffer

States

$S$: State space

$S_t$: System state at time $t$, $S_t \in S$

$S^m_t$: State of the manufacturing system

$S^mic_t$: State of the microgrid system

$S_{env}^t$: State of the environment

$S_{it}^M$: State of machine $i \in \{1, \ldots, N\}$

$S_{it}^B$: State of buffer $i \in \{1, \ldots, N - 1\}$

$g_t^s$: Working status of solar PV at time $t$

$g_t^w$: Working status of wind turbine at time $t$

$g_t^g$: Working status of generator at time $t$

$SOC_i$: State of charge of the battery system at time $t$

$I_t$: Solar irradiance at time $t$

$v_t$: Wind speed at time $t$

General

MDP: Markov Decision Process

TD: Temporal Difference

DPG: Deterministic Policy Gradient

References


