Resumen papper I: "A prescriptive Dirichlet power allocation policy with deep reinforcement Learning"

Código: EL7021-1

Nombre: José Luis Cádiz Sejas

Fuente: Link

Motivo: La temática que se busca para el proyecto final son sistemas prescriptivos (recomendador de acciones), esto con el objetivo de mejorar algún índice que cuantifique costos o producción, por ejemplo, consumo energético, consumo de agua o aumento de producción. Este papper busca optimizar el rendimiento de un sistema de baterías mediante acciones continuas, por lo cual esta muy alineado con lo que se busca en el proyecto final de semestre.

Síntesis:

Este Papper mejora la forma en la que se aprenden las políticas para el caso en el que las acciones son de carácter continuo, para eso se propone una política Dirichet, la cual se compara con una política Gaussiana-softmax que es utilizada usualmente para el caso continuo, pero introduce sesgo durante el entrenamiento.

La contribución de este artículo es que la política Dirichlet proporciona una convergencia más rápida, mejor desempeño y mayor robustez a cambios en los hiperparámetros.

El algoritmo propuesto se usa en un caso de operación prescriptiva en baterías de iones de litio y se demuestra su potencial para mejorar la eficiencia y sostenibilidad de sistemas de múltiples fuentes de energía.

Al comparar los experimentos númericos con ambos enfoques se evidencia la mejora con la distribución de política según Dirichet respecto a las curvas de aprendizaje:

Fig. 2. Numerical experiment results. We compare the learning curves of both output layers with two different learning rates: 0.1 and 0.01, where the shaded areas show the 1-SD confidence intervals over multiple random seeds.

Finalmente, respecto al caso de estudio se evidencia una notable mejora en el sistema de baterías:

Table 2
Average improvement.

Experiments	Average improvement of the working cycle
Four-battery system	15.2%
Eight-battery system	31.9%
Four-second-life-battery system	151.0%

Reliability Engineering and System Safety 224 (2022) 108529

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

A prescriptive Dirichlet power allocation policy with deep reinforcement learning

Yuan Tian [a], Minghao Han [b], Chetan Kulkarni [c], Olga Fink [d,*]

[a] Chair of Intelligent Maintenance Systems, ETH Zürich, Switzerland
[b] Department of Control Science and Engineering, Harbin Institute of Technology, China
[c] KBR, Inc, NASA Ames Research Center, United States
[d] Laboratory of Intelligent Maintenance and Operations Systems, EPFL, Switzerland

ARTICLE INFO

Keywords:
Reinforcement learning
Deep learning
Prescriptive operation
Multi-power source systems

ABSTRACT

Prescribing optimal operation based on the condition of the system, and thereby potentially prolonging its remaining useful lifetime, has tremendous potential in terms of actively managing the availability, maintenance, and costs of complex systems. Reinforcement learning (RL) algorithms are particularly suitable for this type of problem given their learning capabilities. A special case of a prescriptive operation is the power allocation task, which can be considered as a sequential allocation problem whereby the action space is bounded by a simplex constraint. A general continuous action-space solution of such sequential allocation problems has still remained an open research question for RL algorithms. In continuous action space, the standard Gaussian policy applied in reinforcement learning does not support simplex constraints, while the Gaussian-softmax policy introduces a bias during training. In this work, we propose the Dirichlet policy for continuous allocation tasks and analyze the bias and variance of its policy gradients. We demonstrate that the Dirichlet policy is bias-free and provides significantly faster convergence, better performance, and better robustness to hyperparameter changes as compared to the Gaussian-softmax policy. Moreover, we demonstrate the applicability of the proposed algorithm on a prescriptive operation case in which we propose the Dirichlet power allocation policy and evaluate its performance on a case study of a set of multiple lithium-ion (Li-I) battery systems. The experimental results demonstrate the potential to prescribe optimal operation, improving the efficiency and sustainability of multi-power source systems.

1. Introduction

Prescribing an optimal course of action based on the current system state, and thereby potentially prolonging its remaining useful lifetime, has tremendous potential in terms of actively managing the availability, maintenance, and costs of complex systems [1–3]. In fact, it is a sequential decision-making task that either requires very good dynamical models or models with very good learning capabilities. Reinforcement learning (RL) algorithms have recently demonstrated superior performance on sequential decision-making tasks [4]. In particular, model-free RL, which estimates the optimal policy without relying on a model of the dynamics of the environment, has recently yielded very promising results in many challenging tasks across areas as diverse as gaming [5,6], control problems [7,8], prescriptive maintenance [9] and auto machine learning (AutoML) [10].

An important application of prescriptive operations for multi-power source systems is power allocation with the goal of prolonging the lifetime or the usage time of the systems, thereby improving availability, maximizing efficiency, or minimizing cost. These types of prescriptive operation tasks can be considered as sequential allocation problems. One of the major characteristics of allocation problems is that the action space is bounded by a simplex constraint. This constraint makes the application of RL algorithms in a continuous action space particularly challenging. Besides power allocation [11], both sequential and single-step allocation tasks are commonly encountered in several other application scenarios, such as task allocation [12], resource allocation [13,14], order allocation [15], redundancy allocation [16,17] and portfolio management [18]. Particularly for allocation tasks involving complex systems, system state and reliability considerations are crucial.

Several research studies have focused on allocation tasks with reinforcement learning [18–21]. However, one of the main limitations of the previously proposed RL approaches for allocation tasks is that they were solely able to operate in a discretized action space. This discretization typically precludes, on the one hand, fine-grained allocation actions since the number of discretized actions may become

* Corresponding author.
E-mail address: olga.fink@epfl.ch (O. Fink).

https://doi.org/10.1016/j.ress.2022.108529
Received 16 December 2021; Received in revised form 7 April 2022; Accepted 15 April 2022
Available online 30 April 2022

intractably high [22]. On the other hand, the action space needs to be carefully adjusted if the number of possible allocation options changes. These two aspects substantially limit the scalability of the existing approaches.

To enable more general allocation decision-making, continuous action space is required [23,24]. For continuous action-space sequential allocation problems, the RL algorithms need to satisfy the simplex constraints as outlined above. However, the most commonly applied Gaussian policy in other RL tasks is not able to satisfy the simplex constraints [23–26]. Gaussian-softmax policy could be an alternative solution [18]. However, this function is not injective and has additional drawbacks, such as its inability to model multi-modality [27]. This leads to less efficient training and less effective performance.

In this paper, we focus on continuous action-space sequential allocation tasks and propose a Dirichlet-policy-based reinforcement learning framework for sequential allocation tasks. This enables us to overcome the aforementioned limitations. The proposed Dirichlet policy offers several advantages as compared to the Gaussian, Gaussian-softmax, and discretized policies. The Dirichlet policy inherently satisfies the simplex constraint. Moreover, it can be combined with all state-of-the-art stochastic policy RL algorithms. This makes it universally applicable for sequential allocation tasks. Ultimately, the proposed Dirichlet policy exhibits good scalability and transferability properties. In this research, we theoretically demonstrate that the Dirichlet policy is bias-free and results in a lower variance in policy updates as compared to the Gaussian-softmax policy. Finally, we experimentally demonstrate that the Dirichlet policy provides significantly faster convergence, better performance, and is more robust to changes in hyperparameters as compared to the Gaussian-softmax policy.

The performance of the proposed prescriptive operation framework in the context of sequential allocation problems is evaluated on a case study of multi-battery system applications with the goal of prolonging their working cycles. The developed framework only requires raw, real-time current and voltage measurements, along with the incoming power demand, as inputs. To the best of our knowledge, this is the first time an algorithm has been capable of directly performing the load allocation strategy in an end-to-end way (without the involvement of any model-based state estimation). We will demonstrate that, compared to the equally distributed load allocation, the average length of the discharge cycle of the deployed four-battery system can be prolonged by an average of 15.2% (and an eight-battery system by an average of 31.9%) over 5000 random initializations and random load profiles, thereby making the batteries more sustainable. Moreover, we will demonstrate that when implemented on degraded batteries in second-life applications with diverse degradation dynamics, the improvement becomes even more pronounced, reflecting a 151.0% extension of discharge cycles on average and thus enabling the reliable usage of second-life batteries.

The contribution of this paper is twofold: (1) We propose a novel RL-based solution for continuous action-space allocation tasks. In particular, we propose the Dirichlet policy and demonstrate its advantages theoretically and experimentally. (2) Based on the proposed Dirichlet policy, we set forth a prescriptive power allocation framework and evaluate its performance on multi-battery systems to prolong the service cycles of these power source systems. The developed framework shows the potential to improve the efficiency and sustainability of power systems with greater effectiveness.

2. Related work

Prescriptive operation is a comparatively novel research direction that goes beyond merely predicting the evolution of the system condition and the remaining useful life. The main goal of prescriptive operation is to develop algorithms that are not only able to predict the required measures but also to prescribe an optimal course of action based on the current system state. Different objectives can be considered for prescriptive operation tasks, such as prolonging the remaining useful lifetime and thereby improving the reliability and availability of the system; completing a defined mission or reaching an operational goal, even in the case of adverse conditions or faults; and minimizing emissions and energy consumption. Several research studies have recently taken up the concept of prescriptive operation [9,28]. For example, one investigation, taking economic and environmental impact into account, has prescribed an approach to maintenance operation that improves the efficiency of aircraft maintenance [9]. For batteries, optimal charging schedules have been proposed to prolong the remaining useful life (RUL) [29]. Prescriptive operation represents a very promising and urgently required research direction with regard to industrial applications due to the rising complexity and increasingly demanding requirements of complex industrial assets [1,30]. The prescriptive operation problems are, in fact, sequential decision-making problems, for which RL methods have demonstrated very good learning capabilities [9].

In a reinforcement learning task, the agent observes the environment or system state and prescribes an action in order to maximize the cumulative expected future reward. The action space can be discrete, continuous, or mixed. The Q-Learning [31], as well as deep Q-network (DQN) [32] and related variants such as double-DQN [33], are normally designed for discrete action-space tasks. To enable continuous action space, policy-based algorithms such as proximal policy optimization (PPO) [23], trust region policy optimization (TRPO) [26], and soft actor–critic [24] have been proposed. These algorithms represent the stochastic policy via a Gaussian distribution and the agent can sample from the distribution to get the specific action. Besides the stochastic policy, the deep deterministic policy gradient (DDPG) [25] uses a deterministic policy to tackle the continuous action-space problem. However, DDPG produces a relatively weak performance in complex problems [24]. Moreover, beta policy has been proposed to improve the efficiency when physical constraints are present [22].

Allocation tasks are very commonly encountered in real-world prescriptive operation problems. However, the application of reinforcement learning to this type of problem and the elaboration of the theoretical perspective have remained relatively unexplored. The task is to find an optimal distribution of a limited resource given some defined goal and constraints. All allocation tasks need to fulfill the constraint that the action space is bounded by a simplex constraint. Examples of allocation tasks include computational resource allocation, which is highly useful for emerging applications with intense computational resource demands, such as industrial automation [34], blockchain applications [13], and unmanned aerial vehicle (UAV) applications [35]. Reliability redundancy allocation can help improve system reliability and minimize the cost, weight, or volume [36,37]. Order allocation is becoming increasingly important to commercial enterprises like passenger transportation service companies [38,39], food delivery services [40], and other logistics providers [41]. Optimal allocation directly influences the efficiency and profit of such companies, who are relying on limited resources. In the financial field, portfolio management is, in fact, also an allocation problem [18]. Unfortunately, a general solution in RL for allocation problems with the simplex constraint is still lacking and remains an open research question.

A crucial application field of both allocation problems and prescriptive operation is power allocation [42] in multi-power source configurations, which has recently been gaining in importance. A major challenge for power allocation strategies for multi-power source systems has been the design of optimal allocation strategies that take distinct observed states into account and consider different dynamics. For example, in multi-battery systems, the individual batteries commonly start diverging in their states of health and remaining capacities through use [42–44]. Small dissimilarities at the beginning of the lifetime may be amplified by different usage profiles. Once any of the individual batteries reaches the end of discharge (EoD), the normal

operation of the entire system is impacted. Since individual batteries in the system may have dissimilar states of charge that are not directly measurable, distributing the power equally between all batteries is not optimal. Allocating the power demand in an optimal way to each of the individual batteries has the potential to not only prolong the discharge cycle of the entire multi-battery system but also its lifetime, thus improving the sustainability of the batteries.

Different power allocation strategies have been proposed, including rule-based [45–47] and optimization-based approaches [48,49]. There are several limitations to these approaches. In the rule-based load allocation, each specific state would require the definition of customized rules. Thus, the rule-based approaches require extensive prior knowledge as well as extensive experiments for the different conditions that, for example, take into account the state of charge (SoC) or state of health (SoH), which cannot be measured directly. A major drawback of rule-based approaches is that they are typically designed for a specific system and partly for specific usage and operating conditions. Moreover, prior knowledge and model information are also typically required. Therefore, if there is any change in the system configuration or the operating requirements, the allocation rules need to be adjusted. Even for a simple scale-up from four to eight batteries, the allocation rules need to be carefully adjusted. Moreover, since the feedback of such predefined rules is typically delayed, they are also hard to optimize, resulting in sub-optimal solutions.

Optimization-based methods typically require model information. The allocation task is then solved by optimization or control algorithms, such as model predictive control (MPC) [50,51] and Robust MPC [52]. These approaches are vulnerable to uncertainties and changes in the schedule of the power profile. Also, to the best of our knowledge, they all rely on extracted information from physics-based models, such as SoC. Furthermore, they are typically computationally intense, especially for high-dimensional allocation problems. Thus, it is challenging for optimization-based methods to deal with complex real-world systems in real time.

Machine learning approaches have been increasingly applied to different battery management tasks, including predicting the future capacity [53,54], SoC [44,55–57], SoH, and remaining useful life (RUL) [58]. In the power allocation domain, reinforcement learning-based approaches have also been recently investigated in a similar context [19]. Compared to the rule-based and optimization-based approaches mentioned above, the proposed model-free RL-based framework provides an alternative solution while overcoming some of their limitations. Unlike rule-based approaches, the strategies for different systems can be learned with model-free RL without any manual feature engineering or prior knowledge. The learned policy demonstrates superior computational efficiency compared to optimization-based methods. This property is particularly important for real-time applications. Moreover, model-free RL is suitable for finding the optimal policy in tasks where the dynamics are either unknown or affected by a large uncertainty [59]. Under such conditions, the optimization-based methods may fail to find a feasible strategy. Besides, the deep RL typically shows very good performance on end-to-end control tasks and does not require any manual feature engineering. Previous RL-based methodologies addressed power allocation tasks by discretizing the action and state spaces, defining different weight combinations [19,21,60]. This significantly reduces their scalability and transferability. Due to the exploding action space problem, it is not feasible to directly increase the number of weight combinations for a more fine-grid decision-making [25,61]. Thus, to enable a more general power allocation strategy, continuous action space and corresponding approaches [23,24] are needed.

3. Preliminaries

A Markov decision process (MDP) is a discrete-time stochastic control process. At each time step, the process is in a state s_t and its associated agent chooses an action a_t from the set of possible actions. Given the action, the process moves into a new state s_{t+1} at the next step and the agent receives a reward r_t; see Fig. 1 below:

An MDP can be described as a tuple (S, A, r, P, ρ), where S is the set of states that is able to precisely describe the current situation, A is the set of actions, $r(s,a)$ is the reward function, $P(s'|s,a)$ is the transition probability function, and $\rho(s)$ is the initial state distribution.

MDPs have been used to describe an environment in reinforcement learning. In a general reinforcement learning setup, an agent is trained to interact with the environment and get a reward from this interaction. The goal is to find a policy π that maximizes the cumulative reward $J(\pi)$:

$$J(\pi) = \mathbb{E}_{\tau\sim\rho_\pi} \sum_{t=0}^{\infty} r(s_t, a_t) \tag{1}$$

While the standard RL merely maximizes the expected cumulative rewards, the maximum entropy RL framework considers a more general objective [62], which favors stochastic policies. This objective shows a strong connection to the exploration–exploitation trade-off and aims at preventing the policy from getting stuck in local optima. Formally, it is given by:

$$J(\pi) = \mathbb{E}_{\tau\sim\rho_\pi} \sum_{t=0}^{\infty} [r(s_t, a_t) + \beta \mathcal{H}(\pi(\cdot|s_t))], \tag{2}$$

where β is the temperature parameter that controls the stochasticity of the optimal policy.

4. Methodology

To solve the continuous action-space allocation tasks, we introduce for the first time the Dirichlet policy. In the following, we first theoretically demonstrate that the Dirichlet policy is bias-free and has a lower variance of policy update as compared to the Gaussian-softmax policy. Moreover, we experimentally demonstrate that the Dirichlet policy provides a significantly faster convergence, better performance, and is more robust to changes in hyperparameters as compared to the Gaussian-softmax policy. Additionally, we combine the Dirichlet distribution with the state-of-art soft actor–critic for the proposed Dirichlet Power Allocation Policy.

4.1. Implications of the Gaussian policy

In reinforcement learning, a policy is always required to determine which action to take given the current state. In practice, the stochastic policy is usually parameterized by a conditioned Gaussian distribution $\pi_\theta(\mathbf{x}|s) = \mathcal{N}(\mu_\theta(s), \delta_\theta(s))$, where μ and δ are the outputs of the neural networks. However, the action $\mathbf{x}$ sampled from π_θ is not directly applicable to allocation tasks since the constraint $\sum_{i=0}^{N} a_i = 1$ is not satisfied. It is straightforward to pass the generated candidate action $\mathbf{x}$ to a softmax function $\sigma : \mathbb{R}^N \to \mathbb{R}^N$ to obtain the allocation action:

$$a_i = \sigma(x_i)_i = \frac{e^{x_i}}{\sum_{i=1}^{N} e^{x_i}} \tag{3}$$

However, we show in the following that this approach would generate two side effects: a biased estimation and a larger variance. Both of these would jeopardize the policy learning.

4.1.1. Bias

In allocation problems, the policy gradient is written as follows:

$$\mathbb{E}g(\theta) = \mathbb{E}_s \int_0^1 \pi(a|s) \nabla_\theta \log \pi(a|s) Q^\pi(s,a) \mathrm{d}a \tag{4}$$

It should be noted that the softmax function is not injective and many possible $\mathbf{x}$ can result in the same action a. More specifically, the softmax function is invariant under translation by the same value in each coordinate, i.e. $\sigma(\mathbf{x} + c\mathbb{1}) = \sigma(\mathbf{x})$ for any constant $c \in \mathbb{R}$. When

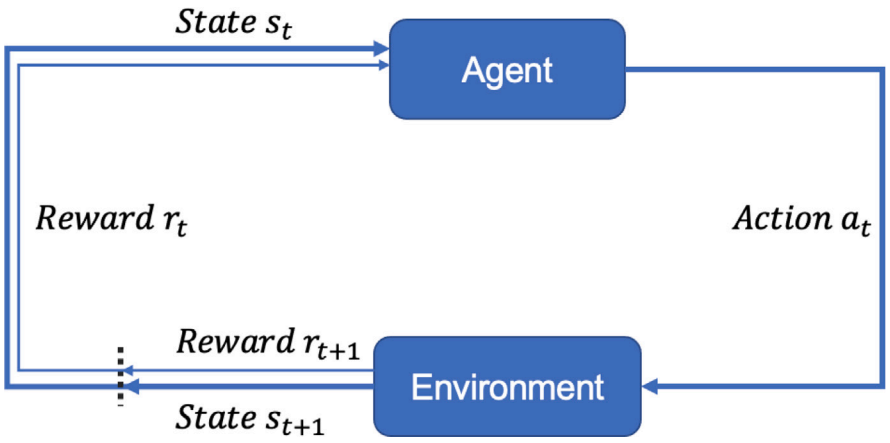

Fig. 1. Reinforcement learning schematic.

the softmax function is combined with the Gaussian policy to generate appropriate allocation actions, the distribution of a is, in fact, relevant to the distribution of the candidate action $\mathbf{x}$ and the probability density function (PDF) satisfies

$$\pi(a|s) = \int_{-\infty}^{\infty} \pi_\theta(\mathbf{x} + c\mathbb{1}|s)\mathrm{d}c \tag{5}$$

Substituting the above relation into the policy gradient follows that

$$\mathbb{E}g(\theta) = \mathbb{E}_s \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \pi_\theta(\mathbf{x} + c\mathbb{1}|s)\nabla_\theta \log \int_{-\infty}^{\infty} \pi_\theta(\mathbf{x} + c\mathbb{1}|s)Q^\pi(s, \sigma(\mathbf{x}))\mathrm{d}\mathbf{x}\mathrm{d}c \tag{6}$$

However, the policy gradient estimator $\mathbb{E}\hat{g}$ used in the ordinary RL algorithm is unaware of the inner integration over the scalar variable c, as in the following

$$\mathbb{E}\hat{g}(\theta) = \mathbb{E}_s \int_{-\infty}^{\infty} \pi_\theta(\mathbf{x}|s)\nabla_\theta \log \pi_\theta(\mathbf{x}|s)Q^\pi(s, \mathbf{x})\mathrm{d}\mathbf{x} \tag{7}$$

As the mapping of the candidate action to the allocation action is done in the environment (the specific allocation task), the estimator is created based on the candidate action and inevitably introduces a bias. Even if we assume that the learned critic based on the candidate action can predict the return precisely, i.e. $Q^\pi(s, \mathbf{x}) = Q^\pi(s, \sigma(\mathbf{x})), \forall x$, the bias still exists due to the unawareness of the marginalization over c.

One might also wonder whether using the transformed allocation action a to compute the policy gradient can yield an unbiased estimation. Unfortunately, this is not the case. This would be equivalent to replacing the candidate action x in (7) with a. Though it looks similar to the form in (4), the distributions π_θ and π are not equivalent. In the end, this will only result in even more biased results.

4.1.2. Variance

In addition to the bias, the Gaussian policy also has the drawback that the variance of the policy gradient estimator is proportional to $1/\sigma^2$. This will induce the variance to reach infinity as the policy converges and becomes deterministic ($\sigma \to 0$) [22].

To illustrate this, a useful insight is gained by comparing the policy gradient with the natural policy gradient [63]. The policy gradient in (7) does not necessarily produce the steepest policy updates [64], while the natural policy gradient does. The natural policy gradient is given by

$$g_{\mathrm{nat}}(\theta) = \mathbb{E}_s F^{-1}(\theta)\hat{g}(\theta) \tag{8}$$

where F denotes the Fisher information matrix, defined as

$$F = \mathbb{E}_{a\sim\pi_\theta}\left[\nabla_\theta \log \pi_\theta(a|s)\nabla_\theta \log \pi_\theta(a|s)^T\right] \tag{9}$$

The policy gradient vector is composed of the length and the direction. The ordinary policy gradient may have the correct direction but not necessarily the correct length. The natural policy gradient adjusts the learning rate according to the policy distribution and produces the steepest step. As shown in [22], the Fisher information matrix for Gaussian policy is proportional to $1/\sigma^2$, which implies that the more deterministic the policy is, the smaller the update step that should be taken. In the end, the constant update steps will overshoot and increase the variance of the policy gradient estimator.

4.2. Dirichlet policy

Since the general Gaussian or Gaussian-softmax policy are not directly applicable to the optimization of allocation problems, applying standard reinforcement learning frameworks or other control algorithms to allocation tasks will result in sub-optimal results that suffer from excessive parameter tuning and/or model complexity. To improve the stability and convergence speed of optimization tasks of allocation problems in continuous action spaces, we propose parameterizing the policy using Dirichlet distribution, which inherently satisfies the simplex constraint and enables an efficient optimization of allocation tasks in continuous action spaces:

$$\pi_\theta(a|s) = \frac{1}{B(\alpha)}\prod_{i=1}^{N} a_i^{\alpha_i - 1} \tag{10}$$

where $B(\alpha)$ denotes the multivariate beta function and can be expressed in terms of the gamma function Γ as follows:

$$B(\alpha) = \frac{\prod_{i=1}^{N}\Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{N}\alpha_i)}. \tag{11}$$

Here, the distribution is shaped by the shape parameters α, which is the output of the neural network $f_\theta(s)$. Thus, the policy is eventually determined by θ.

4.2.1. Bias of the Dirichlet policy

The action a sampled from the Dirichlet policy (10) naturally satisfies the constraints on actions in allocation problems, i.e. $\sum_{i=0}^{N} a_i = 1$ and $a_i \geq 0$ [65]. Thanks to this property, it is possible to directly sample actions that qualify as allocation actions from the Dirichlet policy, without the need to further constrain them. As a result, the policy gradient estimator $\mathbb{E}\hat{g}(\theta)$ for Dirichlet policies takes the same form as the natural policy gradient $\mathbb{E}g(\theta)$ in (4) and is bias-free.

4.2.2. Variance of the Dirichlet policy

Let $A = \sum_{i=1}^{N} \alpha_i$

$$\log \pi_\theta(a|s) = \log(\Gamma(A)) - \sum_{i=1} \log(\Gamma(\alpha_i)) + \sum_{i=1} (\alpha_i - 1)\log(\Gamma(a_i)) \tag{12}$$

Taking the fact that $\partial A/\partial \alpha_i = 1$ and $\partial \alpha_j/\partial \alpha_i = 0$ into account results in:

$$\frac{\partial \log \pi_\theta(a|s)}{\partial \alpha_i} = \psi(A) - \psi(\alpha_i) + \log(a_i) \tag{13}$$

with the second order derivative

$$\frac{\partial^2 \log \pi_\theta(a|s)}{\partial \alpha_i \partial \alpha_j} = \psi'(A) - \psi'(\alpha_i)\delta_{ij} \tag{14}$$

where $\psi'(z) = \psi^{(1)}(z)$ and $\psi^{(m)}(z) = \frac{d^{m+1}}{dz^{m+1}} \ln \Gamma(z)$ is the polygamma function, the m_{th} derivative of the logarithm of the gamma function.

According to the regularity conditions [66], the Fisher information matrix can also be obtained from the second-order partial derivatives of the log-likelihood function,

$$\begin{aligned} F(\alpha) &= -\mathbb{E}_{a\ \pi_\theta} \begin{bmatrix} \frac{\partial^2 \log \pi_\theta(a|s)}{\partial \alpha_1 \partial \alpha_1} & \cdots & \frac{\partial^2 \log \pi_\theta(a|s)}{\partial \alpha_1 \partial \alpha_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 \log \pi_\theta(a|s)}{\partial \alpha_K \partial \alpha_1} & \cdots & \frac{\partial^2 \log \pi_\theta(a|s)}{\partial \alpha_N \partial \alpha_N} \end{bmatrix} \\ &= \begin{bmatrix} \psi'(\alpha_1) - \psi'(A) & \cdots & -\psi'(A) \\ \vdots & \ddots & \vdots \\ -\psi'(A) & \cdots & \psi'(\alpha_N) - \psi'(A) \end{bmatrix} \end{aligned} \tag{15}$$

The variance of the Dirichlet policy is given by $Var[a_i] = \frac{\alpha_i(A-\alpha_i)}{A(A+1)}$. As the policy becomes deterministic given different states, certain allocation actions α_i and A approach infinity simultaneously. As shown in [22], $\psi'(z)$ goes to zero as z goes to infinity. Thus, the inverse of the Fisher information matrix goes to infinity. This ensures that the update steps will not overshoot and the variance of the policy gradient goes to zero.

To summarize, the Dirichlet policy can intrinsically produce unbiased policy gradient estimations, while the variance of policy updates is also guaranteed to be lower than that of the Gaussian policy. These are both favorable properties to enhance the convergence speed and allocation performance.

4.3. Simplex regression experiment

To demonstrate the efficiency and effectiveness of the proposed methodology, we first evaluate it on a simple simplex regression task. The objective is to reconstruct and sequence a 4-dimensional simplex from a 3-dimensional vector obtained by randomly removing a dimension from the target 4-dimensional simplex. For example, given a random simplex vector $[0.4, 0.2, 0.3, 0.1]$, after a dimension is randomly removed, the input data becomes $[0.4, 0.3, 0.1]$. The target output is then the ranked reconstructed simplex $[0.1, 0.2, 0.3, 0.4]$. We use the Mean Average Error (MAE). We apply the proposed Dirichlet policy framework and compare it to the Gaussian-softmax policy. The result shows that the Dirichlet distribution performs better and is more robust to hyperparameters such as, for example, the different learning rates. The Dirichlet policy performs two times better compared to Gaussian-softmax policy with a learning rate of 0.01. In addition, the Dirichlet policy is more robust against different learning rates, while the Gaussian-softmax policy failed with a high learning rate of 0.1. See Fig. 2.

For the neural networks in the numerical experiment, we use a fully connected multi-layer perceptron (MLP) with three hidden layers of 64 units each, outputting the α of a Dirichlet distribution or the μ and σ of a Gaussian distribution. For the Dirichlet distribution network, in the first layer, the Leaky-ReLU activation function [67] is applied. In the second layer, the tanh activation function is applied. The α is modeled by a softplus element-wise operation with $\log(1 + \exp(x))$. A constant 1 is added to the output to make sure that $\alpha \geq 1$. The choice of the activation function is motivated by the design of the Beta policy [22]. For the Gaussian-softmax network, the hidden layers have Leaky-ReLU [67] as the activation function, while the final output layer is mapped to a simplex with a softmax function.

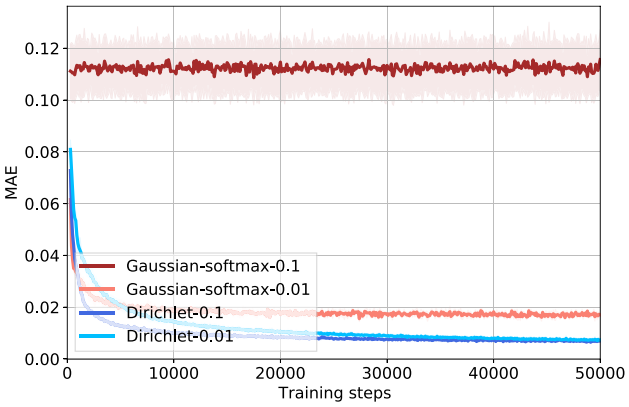

Fig. 2. Numerical experiment results. We compare the learning curves of both output layers with two different learning rates: 0.1 and 0.01, where the shaded areas show the 1-SD confidence intervals over multiple random seeds.

4.4. Soft actor–critic

In this paper, we applied the off-policy reinforcement learning algorithm soft actor–critic (SAC) [24] with the proposed Dirichlet policy. The SAC is based on the maximum entropy reinforcement learning framework [68], where the objective is to maximize both the entropy of the policy and the cumulative return. As a result, it significantly increases training stability and improves exploration during training. Furthermore, it was demonstrated to be 10 to 100 [24] times more data-efficient as compared to any other on-policy algorithms applied to traditional RL tasks.

For the learning of the critic, the objective function is defined as:

$$J(Q) = \mathbb{E}_{(s,a)\sim D}\left[\frac{1}{2}(Q(s,a) - Q_{target}(s,a))^2\right] \tag{16}$$

where Q_{target} is the approximated target of Q:

$$\begin{aligned} &Q_{target}(s,a) = R(s,a) + \\ &\gamma[Q_{target}(s', f(\epsilon, s')) - \beta \log \pi(a'|s')] \end{aligned} \tag{17}$$

The objective function of the policy network is given by:

$$J(\pi) = \mathbb{E}_D\left[\beta[\log(\pi_\theta(f_\theta(\epsilon, s)|s))] - Q(s, f_\theta(\epsilon, s))\right] \tag{18}$$

where π_θ is parameterized by a neural network f_θ, ϵ is an input vector, the $D \doteq \{(s, a, s', r)\}$ is the replay buffer for storing the MDP tuples [32], and β is a positive Lagrange multiplier that controls the relative importance of the policy entropy versus the cumulative return.

4.5. Hyperparameter setting

For the following experiments, we combine the proposed Dirichlet policy with the SAC framework. For the policy network, we use the same architecture design as for the toy experiment with the difference that: (1) 256 units are used and (2) for the additional Q-network, we use a fully connected MLP with three hidden layers of 256 units, outputting the Q-value. All the hidden layers use Leaky-ReLU as the activation function. Fig. 3 illustrates the networks. It is worth pointing out that we adopt the same neural architecture as that used in the SAC [24]. This neural architecture, typically utilized in other control tasks, has yielded good performance in applications including cart–pole balancing [24], humanoid walking [24], real-world robot control [69], real quadrotor control [70], and others [23,26]. More information

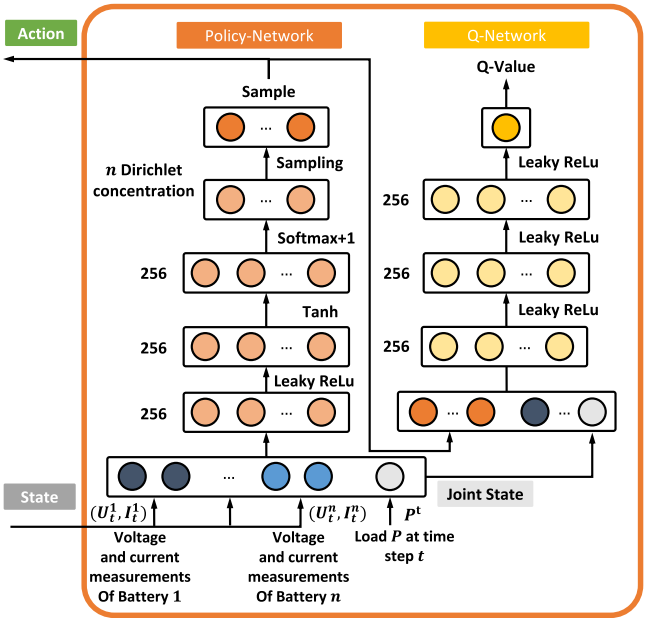

Fig. 3. Overview of the neural network architectures.

Table 1
SAC hyperparameters.

Hyperparameters	Value
Minibatch size	1024
Learning rate - Actor	1e−4
Learning rate - Critic	3e−4
Target entropy	$-\sqrt{d}$
Target smoothing coefficient (τ)	0.005
Discount (γ)	0.99
Updates per step	1

regarding the task-specific input data and output action may be found in Section 5.

Our implementation exploits the double Q-learning technique [71], whereby two Q-functions $\{Q_1, Q_2\}$ are parameterized by neural networks with parameters ν_1 and ν_2. The Q-function with the lower value is exploited in the policy learning step [72], which is useful in mitigating performance degradation caused by the bias in the value estimation.

The optimization of the networks' weights is carried out with the *Adam* algorithm. The *Kaiming* initializer is used for the weight initializations [73]. Table 1 provides a detailed overview of the hyperparameters used for the experiments. Training is conducted on a 2.3 GHz 8-core Intel Core i9 CPU.

5. Power allocation case study

To further evaluate the performance of the proposed method, we design a case study of multi-battery system applications with the goal of prolonging their working cycles. We assume that the power allocation can be controlled at the level of a single battery and that no cell balancing is applied. We would like to emphasize that in this paper, we focus on the algorithm design for the purpose of demonstrating its potential. Implementing this algorithm in real applications would require a dedicated circuit design, which we leave for future work.

The information most commonly used in battery health-related analytics is the operating current and voltage measurements collected by standard battery management systems [44,74]. In this case study, we aim to utilize only raw measurements of current and voltage directly measured on the batteries (before the DC-DC converter) and extend the capability of machine learning from descriptive and predictive analytics to end-to-end prescriptive decision-making. To the best of our knowledge, this is the first time an algorithm has been capable of directly performing the load allocation strategy in an end-to-end manner (without any involvement of model-based state estimation).

The objective of the desired power allocation strategy is to prolong the working cycle of the deployed multi-battery system. To achieve this, we formulate this problem as a Markov decision process (MDP) and propose to solve it with Dirichlet policy reinforcement learning.

Every operation or maneuver of a multi-battery device will impose a power demand P_t on the system. The RL-based strategy will prescribe an action a_t that dynamically allocates the power demand P_t based on the observed state s_t. In our case, s_t is represented by the real-time operational current and voltage of all the batteries in the system and the total power demand, resulting in $s_t = [V_t, I_t, P_t]$. Then, the system's state changes according to the allocation strategy and the system dynamics $\mathcal{P}$. To achieve the objective of prolonging the working cycle of the battery device, we provide a reward of $r_t = 1$ to the agent at each time step at which all the batteries in the system are still operational or the voltages are all higher than the end-of-discharge (EoD) state. Given a discount factor $\gamma \leq 1$, an optimal allocation strategy maximizes the expected discounted sum of future rewards, or return:

$$R_\tau = \mathbb{E}[\sum_{t=0}^{\inf} \gamma^t r(s_t, a_t) | s_0 = s] \tag{19}$$

where $\mathbb{E}$ indicates the expected value. R_s characterizes the long-term value of the allocation strategy from an initial state s_0 onwards.

Fig. 4 provides an overview of the Dirichlet power allocation framework. **A:** When deploying the proposed strategy on any device with a multi-battery system – such as a quadrotor, a robot, or an electric car – any maneuver induces a load demand. For every maneuver, the trained strategy receives the incoming load demand with the real-time current–voltage measurement. It distributes the power based only on the received information or observation, without any online optimization. **B:** The proposed strategy is represented by a neural network, which takes the measurements as input and outputs a weight combination on how the load should be distributed to the individual batteries. The trained network can dynamically allocate the power in an end-to-end way without any estimation of the degradation state. With the input information of current and voltage measurements, it can first implicitly learn the health of the batteries – such as SoC, SoH, or RUL – for decision-making. With the proposed Dirichlet policy, which inherently satisfies the simplex constraint of the allocation tasks, it can prescribe fine-grid allocation weights in a continuous manner and can be trained more efficiently and effectively. **C:** In this paper, the objective is prolonging the working cycle of the deployed multi-battery systems, a goal which could be changed in other tasks according to different requirements.

5.1. Simulation environment

We train the allocation strategy in a simulation environment. The simulation environment is a multiple Li-I battery system computational model from the NASA prognostic model library [75,76]. It captures the relevant electrochemical processes of the discharge. For an individual battery, the state changes over time as a function of input load and current system states are given by:

$$\begin{aligned} x(k+1) &= f(k, x(k), \theta(k), u(k)), \\ y(k+1) &= g(\mathbf{x}_{t+1}, \theta(k), u(k), n(k)), \end{aligned} \tag{20}$$

where k is a discrete time variable, $x(k) \in \mathbb{R}^{n_x}$ is a state vector, $\theta(k) \in \mathbb{R}^{n_\theta}$ is an unknown parameter vector, $u(k) \in \mathbb{R}^{n_u}$ is the input vector, f is the state equation, $y(k) \in \mathbb{R}^{n_y}$ is the output vector, and h is the output equation. For more details on the battery model, we refer interested readers to the original paper [75].

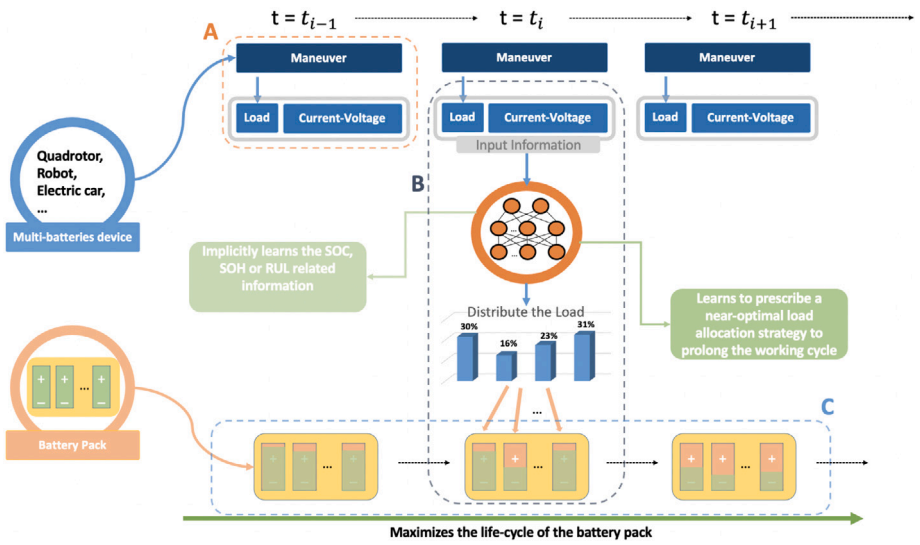

Fig. 4. Overview of the power allocation for multi-battery systems.

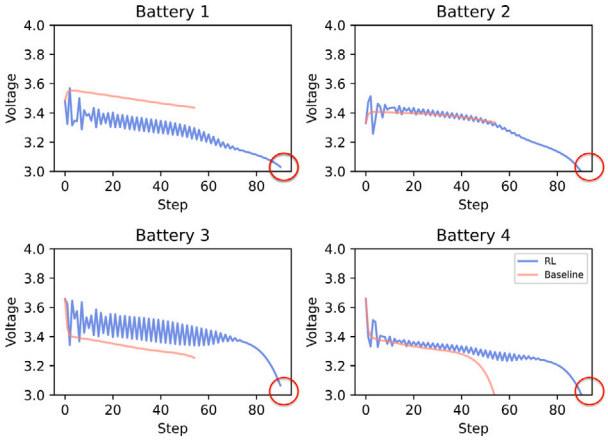

Fig. 5. Discharge trajectories of a randomly selected test case: The y-axis represents the observed operational voltage of the corresponding batteries. The x-axis represents the decision-making steps.

During the discharge process, the load is allocated at the level of a single battery cell and no load balancing is performed. This computational model serves as a reliable proxy for actual battery dynamics and allows for fast iterations over the controller design. It is worth mentioning that batteries generally have relatively complex working dynamics, which is also a challenging case study by which to evaluate the general performance of a power allocation strategy.

For training, we randomly sample battery states during operation as initial states s_0 for any new episode. The episode will be re-initialized when any of the batteries reaches the EoD state.

5.2. Results

The proposed framework is evaluated with respect to three performance aspects: (1) Performance is assessed on a battery system consisting of four Li-I cells. (2) Scalability is evaluated on a battery system consisting of eight Li-I cells. (3) Transferablity is evaluated on a battery system consisting of four second-life Li-I cells, where each of the batteries exhibits different degradation dynamics. (4) We compare the learning performance to other state-of-the-art reinforcement learning methods, (5) and also to heuristic strategies. We summarize the average improvement $\frac{\text{total steps(ours)-total steps(baseline)}}{\text{total steps(baseline)}}$ over the baseline; see Table 2. All the performance metrics are averaged among 5000 different random initializations with random load profiles.

Table 2
Average improvement.

Experiments	Average improvement of the working cycle
Four-battery system	15.2%
Eight-battery system	31.9%
Four-second-life-battery system	151.0%

(1) Performance evaluation on a four-Li-I-battery system. The trained strategy is tested on 5000 different random initializations with random load profiles. Compared to the baseline strategy (distributing the power equally between all batteries), the proposed framework prolongs the working cycle by 15.2% on average. We can observe that the single batteries were controlled by the RL algorithm in such a way that they tended to reach the EoD state at approximately the same time (Fig. 5). This is an indication of near-optimal performance. The proposed strategy also demonstrates a relatively smooth allocation profile (Fig. 5).

(2) Scalability evaluation on an eight-Li-I battery system. Scalability is an essential requirement for power allocation approaches since different assets will have different numbers of configurations. Previous RL approaches discretize the action and state spaces, defining different weight combinations [19,21], which needs to redesign the action space when scaling up the system size. We present the proposed approach on an eight-battery system, following the same setup as for the system with four batteries, and show good scalability. Since more batteries provide more flexibility, the proposed RL framework again displays superior performance as compared to the baseline. The performance improvement is significantly higher when compared to the four-battery case study: Over all the test cases, the lifetime can be extended by 31.9% on average in comparison to the baseline. Similar properties can be observed as in the four-battery case: The batteries can reach the EoD state nearly simultaneously (Fig. 6), indicating a near-optimal allocation performance. The discharge curves are partly influenced by the allocation strategy. The oscillation represents the changing weights for the load allocated to each of the batteries. We observed on the performed experiments that the RL policy appears to prefer frequently changing the weights between the different batteries to prolong the working cycle. The investigation of this behavior and the improvement of the interpretability will inform our future work.

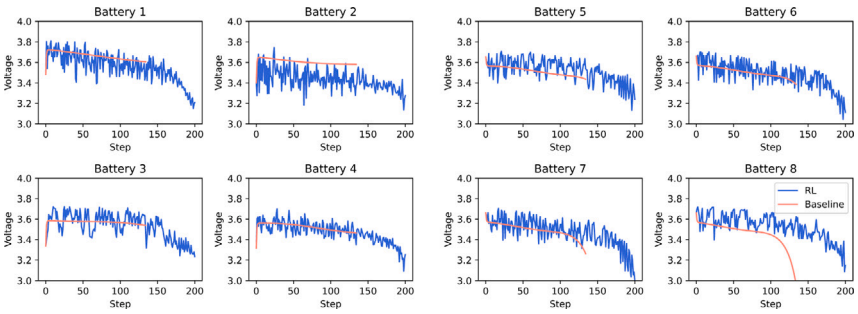

Fig. 6. Eight-battery system case result. A randomly selected set of discharge trajectories from the test cases. The *y*-axis represents the observed operational voltage of the corresponding batteries, while the *x*-axis represents the decision-making steps.

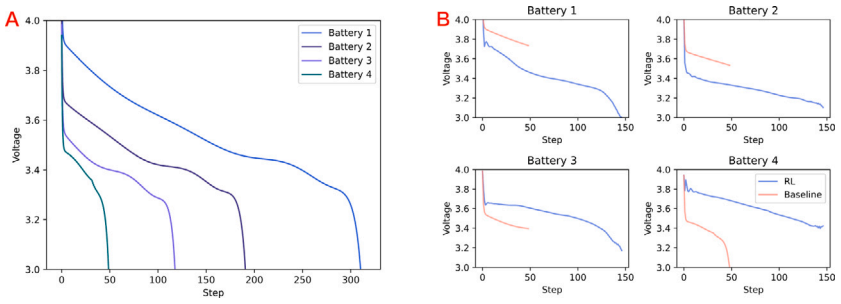

Fig. 7. Second-life battery system case result. The *y*-axis represents the observed operational voltage of the corresponding batteries, while the *x*-axis represents the decision-making steps.

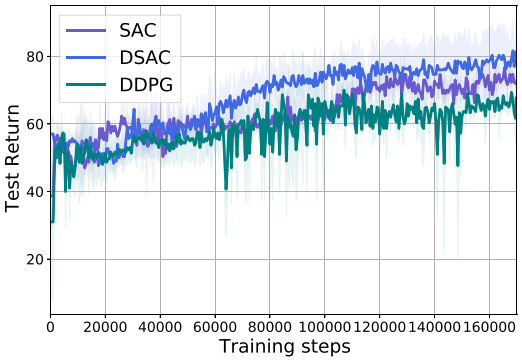

Fig. 8. Learning performance and reproducibility, where the shaded areas show the standard deviation confidence intervals over three random seeds. The *x*-axis indicates the total time steps. The *y*-axis indicates the test return.

(3) Transferablity evaluation based on a four-second-life Li-I battery system. In this research, to evaluate the transferability of the proposed approach to systems with different degradation dynamics [77], we consider batteries in second-life applications [42,78,79]. Even under the same state initialization and same load profile, second-life batteries with dissimilar degradation dynamics will reach the EoD state much earlier. In Fig. 7, **A** presents the voltage trajectories of four batteries. From battery 1 to 4, the degradation becomes more notable. Even under the same initialization and same load profile, the discharge curve changes significantly. **B** is a randomly chosen trajectory. The policy could significantly prolong the working cycle of the deployed second-life battery cases.

For this evaluation, we keep all settings similar to those from the previous two experiments. On average, the proposed approach achieves a 151.0% improvement as compared to the equal load distribution.

The proposed approach demonstrates even more potential in systems with different power source dynamics or degraded assets.

(4) Learning performance compared to the state of the art (SOTA). To further evaluate the performance and reproducibility of the results of the proposed Dirichlet policy, we compare it to two alternative RL algorithms: the original SAC [24], one of the state-of-the art reinforcement learning algorithms, and the deep deterministic policy gradient (DDPG) [25]. We train all the agents over three different seeds on the four-battery case study. As shown in Fig. 8, we observe that the proposed Dirichlet-SAC (DSAC) exhibits considerable reproducibility along with superior performance and convergence speed compared to the original SAC and the DDPG.

(5) Comparison to heuristic strategies To the best of our knowledge, there are no optimization-based approaches allowing solely on voltage–current measurements. We perform this comparison for the sake of completeness of the evaluations. We would also like to emphasize that this is not a fair comparison. Since we would like to prolong the time to the EoD state, we define four heuristic strategies based on the operation voltage. We compare the average relative performance $\frac{\text{working cycle}}{\text{baseline working cycle}}$ to the baseline among 5000 different random initializations with random load profiles. However, these strategies actually yield inferior performance as compared to the baseline, see Table 3.

The weights in the Table 3 means to distribute the weighted load to the batteries with voltages from low to high, respectively.

6. Conclusion

In this work, a novel prescriptive Dirichlet policy reinforcement learning framework is proposed for continuous allocation tasks. The proposed method overcomes the bias estimation and large variance problems in policy gradient and can be applied to any general real-world allocation task. It is also compatible with all other continuous control reinforcement learning algorithms with stochastic policies. In addition, for a specific real-world prescriptive operation task, the power allocation task, we introduce the Dirichlet power allocation policy,

Table 3
Performance comparison to heuristic rules.

Approaches	Weights	Relative performance
Rule I	[0.15, 0.25, 0.25, 0.35]	0.758
Rule II	[0.1, 0.2, 0.3, 0.4]	0.279
Rule III	[0.1, 0.2, 0.2, 0.5]	0.076
Rule IV	[0.05, 0.2, 0.35, 0.45]	0.120
Proposed method	Learned	1.152

which presents an effective and data-based prescriptive framework that is fully autonomous, flexible, transferable, and scalable. The developed framework has the potential to improve the efficiency and sustainability of multi-power source systems. To the best of our knowledge, it is also the first framework that enables distribution of the load in an end-to-end learning setup, without any additional inputs of, e.g., SoC estimation. In future work, we aim to apply and deploy the proposed framework to more challenging and extensive real-world power allocation tasks and extend it for larger problems in order to evaluate its limitations.

CRediT authorship contribution statement

Yuan Tian: Conceptualization, Methodology, Writing – review & editing, Writing – original draft, Visualization, Validation, Formal analysis. **Minghao Han:** Methodology, Writing – original draft. **Chetan Kulkarni:** Writing – review & editing. **Olga Fink:** Conceptualization, Writing – review & editing, Writing – original draft, Supervision, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the Swiss National Science Foundation under Grant PP00P2_176878. The work by Chetan Kulkarni was supported under NASA Ames Research Center, Contract No. 80ARC020D0010.

References

[1] Ansari F, Glawar R, Nemeth T. PriMa: a prescriptive maintenance model for cyber-physical production systems. Int J Comput Integr Manuf 2019;32(4–5):482–503.
[2] Ansari F, Glawar R, Sihn W. Prescriptive maintenance of CPPS by integrating multimodal data with dynamic bayesian networks. In: Machine learning for cyber physical systems, technologies for intelligent automation. Vol. 11. 2020, p. 1–8.
[3] Popp T, Shirangi MG, Nipen OP, Berggreen A. Prescriptive data analytics to optimize casing exits. In: IADC/SPE international drilling conference and exhibition. OnePetro; 2020.
[4] Sutton RS, Barto AG, Williams RJ. Reinforcement learning is direct adaptive optimal control. IEEE Control Syst Mag 1992;12(2):19–22.
[5] Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, et al. Playing atari with deep reinforcement learning. 2013, arXiv preprint arXiv:1312.5602.
[6] Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M. Deterministic policy gradient algorithms. In: Proceedings of the 31st international conference on international conference on machine learning. Vol. 32. 2014, p. I–387.
[7] Han M, Tian Y, Zhang L, Wang J, Pan W. H infinity model-free reinforcement learning with robust stability guarantee. 2019, arXiv preprint arXiv:1911.02875.
[8] Tian Y, Chao MA, Kulkarni C, Goebel K, Fink O. Real-time model calibration with deep reinforcement learning. Mech Syst Signal Process 2022;165:108284.
[9] Meissner R, Rahn A, Wicke K. Developing prescriptive maintenance strategies in the aviation industry based on a discrete-event simulation framework for post-prognostics decision making. Reliab Eng Syst Saf 2021;107812.
[10] Tian Y, Wang Q, Huang Z, Li W, Dai D, Yang M, et al. Off-policy reinforcement learning for efficient and effective gan architecture search. In: European conference on computer vision. Springer; 2020, p. 175–92.
[11] Zhang L, Zheng H, Wan T, Shi D, Lyu L, Cai G. An integrated control algorithm of power distribution for islanded microgrid based on improved virtual synchronous generator. IET Renew Power Gener 2021.
[12] Deng X, Li J, Liu E, Zhang H. Task allocation algorithm and optimization model on edge collaboration. J Syst Archit 2020;110:101778.
[13] Feng J, Yu FR, Pei Q, Du J, Zhu L. Joint optimization of radio and computational resources allocation in blockchain-enabled mobile edge computing systems. IEEE Trans Wireless Commun 2020;19(6):4321–34.
[14] Zhang X, Ding S, Ge B, Xia B, Pedrycz W. Resource allocation among multiple targets for a defender-attacker game with false targets consideration. Reliab Eng Syst Saf 2021;211:107617.
[15] Feng J, Gong Z. Integrated linguistic entropy weight method and multi-objective programming model for supplier selection and order allocation in a circular economy: A case study. J Cleaner Prod 2020;277:122597.
[16] Zhang H, Li Y-F. Robust optimization on redundancy allocation problems in multi-state and continuous-state series-parallel systems. Reliab Eng Syst Saf 2021;108134.
[17] Nath R, Muhuri PK. Evolutionary optimization based solution approaches for many objective reliability-redundancy allocation problem. Reliab Eng Syst Saf 2021;108190.
[18] Jiang Z, Xu D, Liang J. A deep reinforcement learning framework for the financial portfolio management problem. 2017, arXiv preprint arXiv:1706.10059.
[19] Xiong R, Cao J, Yu Q. Reinforcement learning-based real-time power management for hybrid energy storage system in the plug-in hybrid electric vehicle. Appl Energy 2018;211:538–48.
[20] Yang T, Hu Y, Gursoy MC, Schmeink A, Mathar R. Deep reinforcement learning based resource allocation in low latency edge computing networks. In: 2018 15th International symposium on wireless communication systems. IEEE; 2018, p. 1–5.
[21] Maia R, Mendes J, Araújo R, Silva M, Nunes U. Regenerative braking system modeling by fuzzy Q-learning. Eng Appl Artif Intell 2020;93:103712.
[22] Chou P-W, Maturana D, Scherer S. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In: International conference on machine learning. PMLR; 2017, p. 834–43.
[23] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. 2017, arXiv preprint arXiv:1707.06347.
[24] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: International conference on machine learning. PMLR; 2018, p. 1861–70.
[25] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning. 2015, arXiv preprint arXiv:1509.02971.
[26] Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. Trust region policy optimization. In: International conference on machine learning. PMLR; 2015, p. 1889–97.
[27] Joo W, Lee W, Park S, Moon I-C. Dirichlet variational autoencoder. Pattern Recognit 2020;107:107514.
[28] Consilvio A, Sanetti P, Anguìta D, Crovetto C, Dambra C, Oneto L, et al. Prescriptive maintenance of railway infrastructure: From data analytics to decision support. In: 2019 6th International conference on models and technologies for intelligent transportation systems. IEEE; 2019, p. 1–10.
[29] Sui Y, Song S. A multi-agent reinforcement learning framework for lithium-ion battery scheduling problems. Energies 2020;13(8):1982.
[30] Vater J, Harscheidt L, Knoll A. Smart manufacturing with prescriptive analytics. In: 2019 8th International conference on industrial technology and management. IEEE; 2019, p. 224–8.
[31] Watkins CJ, Dayan P. Q-learning. Mach Learn 1992;8(3–4):279–92.
[32] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature 2015;518(7540):529–33.
[33] Hasselt H. Double Q-learning. Adv Neural Inf Process Syst 2010;23:2613–21.
[34] Chen Y, Li Z, Yang B, Nai K, Li K. A stackelberg game approach to multiple resources allocation and pricing in mobile edge computing. Future Gener Comput Syst 2020;108:273–87.
[35] Shimada H, Kawamoto Y, Kato N. Novel computation and communication resources allocation using relay communications in UAV-mounted cloudlet systems. IEEE Trans Netw Sci Eng 2021.
[36] Wang W, Lin M, Fu Y, Luo X, Chen H. Multi-objective optimization of reliability-redundancy allocation problem for multi-type production systems considering redundancy strategies. Reliab Eng Syst Saf 2020;193:106681.
[37] Sabri-Laghaie K, Karimi-Nasab M. Random search algorithms for redundancy allocation problem of a queuing system with maintenance considerations. Reliab Eng Syst Saf 2019;185:144–62.
[38] Kamandanipour K, Nasiri MM, Konur D, Yakhchali SH. Stochastic data-driven optimization for multi-class dynamic pricing and capacity allocation in the passenger railroad transportation. Expert Syst Appl 2020;158:113568.
[39] Cao C, Feng Z. Optimal capacity allocation under random passenger demands in the high-speed rail network. Eng Appl Artif Intell 2020;88:103363.

[40] Sun G, Tian Z, Liu R, Jing Y, Ma Y. Research on coordination and optimization of order allocation and delivery route planning in take-out system. Math Probl Eng 2020;2020.
[41] Jauhar SK, Amin SH, Zolfagharinia H. A proposed method for third-party reverse logistics partner selection and order allocation in the cellphone industry. Comput Ind Eng 2021;162:107719.
[42] Hu X, Xu L, Lin X, Pecht M. Battery lifetime prognostics. Joule 2020;4(2):310–46.
[43] Zheng F, Jiang J, Zaidan MA, He W, Pecht M. Prognostics of lithium-ion batteries using a deterministic Bayesian approach. In: 2015 IEEE conference on prognostics and health management. IEEE; 2015, p. 1–4.
[44] Severson KA, Attia PM, Jin N, Perkins N, Jiang B, Yang Z, et al. Data-driven prediction of battery cycle life before capacity degradation. Nat Energy 2019;4(5):383–91.
[45] Wang Y, Zeng X, Song D, Yang N. Optimal rule design methodology for energy management strategy of a power-split hybrid electric bus. Energy 2019;185:1086–99.
[46] Wang Y, Sun Z, Chen Z. Development of energy management system based on a rule-based power distribution strategy for hybrid power sources. Energy 2019;175:1055–66.
[47] Leonori S, Paschero M, Mascioli FMF, Rizzi A. Optimization strategies for microgrid energy management systems by genetic algorithms. Appl Soft Comput 2020;86:105903.
[48] Bai Y, He H, Li J, Li S, Wang Y-x, Yang Q. Battery anti-aging control for a plug-in hybrid electric vehicle with a hierarchical optimization energy management strategy. J Cleaner Prod 2019;237:117841.
[49] Zhang S, Xiong R, Cao J. Battery durability and longevity based power management for plug-in hybrid electric vehicle with hybrid energy storage system. Appl Energy 2016;179:316–28.
[50] Ishii K. MPC based power allocation for reliable wireless networked control systems. IEEE Access 2021;9:60913–22.
[51] Chen H, Chen J, Lu H, Yan C, Liu Z. A modified MPC-based optimal strategy of power management for fuel cell hybrid vehicles. IEEE/ASME Trans Mechatronics 2020;25(4):2009–18.
[52] Huang Y, Wang H, Khajepour A, He H, Ji J. Model predictive control power management strategies for HEVs: A review. J Power Sources 2017;341:91–106.
[53] Nagulapati VM, Lee H, Jung D, Brigljevic B, Choi Y, Lim H. Capacity estimation of batteries: Influence of training dataset size and diversity on data driven prognostic models. Reliab Eng Syst Saf 2021;216:108048.
[54] Yang D, Zhang X, Pan R, Wang Y, Chen Z. A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. J Power Sources 2018;384:387–95.
[55] Liu X, Zheng Z, Büyüktahtakın İE, Zhou Z, Wang P. Battery asset management with cycle life prognosis. Reliab Eng Syst Saf 2021;216:107948.
[56] Ng M-F, Zhao J, Yan Q, Conduit GJ, Seh ZW. Predicting the state of charge and health of batteries using data-driven machine learning. Nat Mach Intell 2020;1–10.
[57] Jiao M, Wang D, Yang Y, Liu F. More intelligent and robust estimation of battery state-of-charge with an improved regularized extreme learning machine. Eng Appl Artif Intell 2021;104:104407.
[58] Xu X, Tang S, Yu C, Xie J, Han X, Ouyang M. Remaining useful life prediction of lithium-ion batteries based on Wiener process under time-varying temperature condition. Reliab Eng Syst Saf 2021;214:107675.
[59] Buşoniu L, de Bruin T, Tolić D, Kober J, Palunko I. Reinforcement learning for control: Performance, stability, and deep approximators. Annu Rev Control 2018.
[60] Xu Y, Pi D, Yang S, Chen Y. A novel discrete bat algorithm for heterogeneous redundancy allocation of multi-state systems subject to probabilistic common-cause failure. Reliab Eng Syst Saf 2021;208:107338.
[61] Bellman R. Dynamic programming and Lagrange multipliers. Proc Natl Acad Sci USA 1956;42(10):767.
[62] Ziebart BD. Modeling purposeful adaptive behavior with the principle of maximum causal entropy. 2010.
[63] Kakade SM. A natural policy gradient. Adv Neural Inf Process Syst 2001;14:1531–8.
[64] Amari S-I. Natural gradient works efficiently in learning. Neural Comput 1998;10(2):251–76.
[65] Kotz S, Balakrishnan N, Johnson NL. Continuous multivariate distributions, volume 1: models and applications. Vol. 1. John Wiley & Sons; 2004.
[66] Wasserman L. All of statistics: a concise course in statistical inference. Springer Science & Business Media; 2013.
[67] Maas AL, Hannun AY, Ng AY, et al. Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. Vol. 30. No. 1. Citeseer; 2013, p. 3.
[68] Haarnoja T, Tang H, Abbeel P, Levine S. Reinforcement learning with deep energy-based policies. 2017, arXiv preprint arXiv:1702.08165.
[69] Mahmood AR, Korenkevych D, Komer BJ, Bergstra J. Setting up a reinforcement learning task with a real-world robot. In: 2018 IEEE/RSJ international conference on intelligent robots and systems. IEEE; 2018, p. 4635–40.
[70] Hwangbo J, Sa I, Siegwart R, Hutter M. Control of a quadrotor with reinforcement learning. IEEE Robot Autom Lett 2017;2(4):2096–103.
[71] Van Hasselt H, Guez A, Silver D. Deep reinforcement learning with double q-learning. In: Thirtieth AAAI conference on artificial intelligence. 2016.
[72] Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: International conference on machine learning. 2018, p. 1587–96.
[73] He K, Zhang X, Ren S, Sun J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. 2015, p. 1026–34.
[74] Richardson RR, Birkl CR, Osborne MA, Howey DA. Gaussian process regression for in situ capacity estimation of lithium-ion batteries. IEEE Trans Ind Inf 2018;15(1):127–38.
[75] Daigle MJ, Kulkarni CS. Electrochemistry-based battery modeling for prognostics. 2013.
[76] https://github.com/nasa/PrognosticsModelLibrary.
[77] Chao MA, Kulkarni C, Goebel K, Fink O. Fusing physics-based and deep learning models for prognostics. Reliab Eng Syst Saf 2022;217:107961.
[78] Peterson SB, Whitacre J, Apt J. The economics of using plug-in hybrid electric vehicle battery packs for grid storage. J Power Sources 2010;195(8):2377–84.
[79] Fink O, Wang Q, Svensen M, Dersin P, Lee W-J, Ducoffe M. Potential, challenges and future directions for deep learning in prognostics and health management applications. Eng Appl Artif Intell 2020;92:103678.