Practice What You Preach: One Shot Optimization versus Sequential Decision Problems

Inno Zhai

Abstract

This note highlights common mistakes in the problem formulation of communication system works that involve resource allocation and reinforcement learning (RL). Specifically, the main mistake of these works is that they formulate a deterministic or one shot optimization problem and then proceed to solve it via an RL approach. As result, these works are misleading, and should be read with care. This note also provides example works with an incorrect and correct problem formulation.

1 Motivation

A standard structure in many papers that aim to optimize resources in communication systems is to first outline a system model, formulate an optimization problem and then propose a solution. Many of these papers, however, present a one-shot or deterministic optimization problem and then proceed to solve it using a reinforcement learning (RL) approach; see Section 4. Unfortunately, RL solves sequential stochastic decision problems. Its main aim is to determine a *policy* that optimizes an *expected* reward. That is, given a state, the policy returns an action that optimizes the *expected* cumulative reward; hence, a policy is a function.

Apart from that, many works also suffer from the following errors:

- They formulate an optimization problem *without* considering uncertainties but yet they solve a problem *with* uncertainties, e.g., [1]. In other words, they formulate a deterministic resource allocation problem but then proceed to use RL to optimize an expected reward.
- Many works *reformulate* their formulated problem, e.g., [2], [3], turning a deterministic one-shot optimization problem into a sequential decision problem; both of which are fundamentally different problems. However, reformulation means the new problem is *equivalent* to the original problem in the sense that a solution for the reformulated problem is also a solution for the original problem.
- RL involves making decisions over different stages such as time. However, the problem formulation in many works does not even model this fact; i.e., they do not consider decision making over time.

Next, this note aims to highlight the differences between deterministic optimization and sequential decision problems, where it first provides the necessary background of both types of problems before discussing their differences. After that, it lists papers that contain an incorrect and correct problem formulation.

2 Background

2.1 Deterministic Optimization Problems

An optimization problem consists of the following ingredients [4], [5]:

- Objective function $f(x)$. This quantifies the performance of a decision or solution.
- Decision variables (x). These are a set of values or system quantities that we would like to determine.
- Constraints ($g(x) \leqslant 0$). These constraints determine the values that decision variables are allowed to take.

An optimization problem can be written as follows:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & Ax = b, \end{aligned} \tag{1}$$

where A is a matrix of coefficients and b is a vector that bounds the amount of resources. In plain English, the aim is to find an x value or a vector $\mathbf{x}$ that yields the minimal $f(x)$ value; note, the problem can be transformed into a maximization problem by optimizing $-f(x)$ instead.

Many works involve problems that span multiple time slots; $\mathcal{T} = \{1, 2, \ldots, T\}$. An example is

$$\begin{aligned} \underset{x^t}{\text{minimize}} \quad & \sum_{t \in \mathcal{T}} c^t x^t \\ \text{subject to} \quad & g_i(\mathbf{x}) = 0, i = 1, \ldots, M, \end{aligned} \tag{2}$$

where M is the number of constraints, and $\mathbf{x} = \{x^1, x^2, \ldots, x^T\}$. The previous formulation may include a constraint such as $I^t = I^{t-1} - ax^t + d^t y^t$; for example, I^t may indicate the amount of energy in an electric vehicle (EV)'s battery at time t, where it is a function of its previous energy level I^{t-1}, usage x^t and charging duration y^t.

For example problems involving multiple periods or time slots, please look up (i) multi-period inventory optimization problems, and (ii) multi-period portfolio optimization problems; both are classic problems that aim to optimize resources and decisions that are coupled across time.

The aforementioned problems have the following key characteristics:

- For problems involving time or different stages, all information is known upfront. However, this is not practical because they require causal or future information; having said that, they are useful as theoretical benchmarks. Note that the issue with non-causal information can be addressed using receding horizon control or model predictive control [6] or via stochastic programming [7].
- There are no random variables. This is the meaning of *deterministic*, where all the coefficient of variables and resource bounds are given as inputs. Hence, it does not not involve averages or expectation of random variables. Readers interested in dealing with random coefficients are referred to the literature on stochastic optimization [7].
- The computed solution x is only optimal for coefficients A and b. This means if any of these coefficients changed, the solution may have to change, requiring an optimization problem to be recomputed using new coefficient values. In this respect, they are *one-shot* optimization problems; i.e., given the coefficient of variables, they are solved *once* to yield the optimal solution or objective function value.

2.2 Sequential Decision Processes

The objective of RL is to determine the optimal *policy* π^* that maximizes an expected reward [8]–[10]. Formally,

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r^t \right], \tag{3}$$

where $\gamma^t \in [0, 1]$ is a discount factor and r^t is the reward obtained at time t. In other words, we aim to find a policy that helps an agent makes a decision in each time t where the sum of discounted reward is maximized; note, we can also choose to minimize rewards. Further, the expectation is taken over different sample paths or random realizations of the system when using policy π.

A sequential decision (stochastic) problem has five elements [11]:

- **State** S^t. This could model the load of a server or the gain of a wireless channel.
- **Action** a or $a = \pi(S^t)$. This could denote a decision to offload a task, deploy vehicles, the transmit power of devices to name a few.
- **Exogenous information** W^t. This is the information revealed by the system/environment at time t. For example, it could represent the position of vehicles or task arrivals at servers.
- **Transition function**:

$$S^{t+1} = S^M(S^t, \pi(S^t), W^{t+1}), \tag{4}$$

where $S^M(.)$ represents the system/plant model that governs system evolution given system state S^t, action $\pi(S^t)$ and exogenous information W^{t+1}.

- **Objective function** $C(S^t, \pi(S^t))$. This is the cost/reward received by an agent for taking action $\pi(S^t)$.

The goal is then to find a policy π from the set of policies, denoted as Π, that minimizes the expected cost. Mathematically, we have

$$\min_{\pi \in \Pi} \mathbb{E}^{\pi} \sum_{t=0}^{T} C(S^t, \pi(S^t)), \tag{5}$$

where $S^{t+1} = S^M(S^t, \pi(S^t), W^{t+1})$, and the expectation is taken over all possible realizations of $W^1, W^2, \ldots, W^T$ when using policy π. Note, $S^M(.)$ represents a transition model of the system.

A sequential decision process has the following key characteristics:

- It models an agent that takes actions over time or different stages to optimize an expected reward. The agent uses a *policy* or a function to act or make decisions based on system states.
- It models a system that evolves over time, namely $S^{t+1} = S^M(S^t, \pi(S^t), W^{t+1})$, where a system's *state* changes upon taking an action or due to exogenous information. In addition, each action yields a reward, and an agent may receive a reward that is a function of state changes or governed by a stochastic process.

3 KEY DIFFERENCES

There are a number of fundamental differences between deterministic optimization (DO) and sequential decision (SD) problems. To elaborate:

1) SD problems involve making decisions over time using policy π. For each decision, there is a corresponding cost/reward. Further, each decision is made using non-causal information, and an agent aims to optimize its average cumulative reward. DO problems, however, do not have this aim.
2) DO problems that consider multiple periods assume future information is available upfront; this is also called non-causal information. Given this information, solving a DO problem reveals the decision over all time slots. By contrast, SD problems are solved iteratively over time. Further, in SD problems, once a decision/action is taken in time slot t, it cannot be changed in a future time slot. On the other hand, for DO problems with multiple periods, as they have future information, they are able to adjust the decisions in previous time slots. Clearly, this is not practical.
3) SD problems often involve random sample paths, i.e., sequences of random variables that are governed by the environment and/or actions taken by an agent. However, DO problems have a fixed/deterministic system or resource state. This means if there is a change in the parameters (coefficients or cost) of a DO problem, then the resulting solution may not be optimal or may become infeasible. On the other hand, for SD problems, the computed policy works across different scenarios or sample paths.
4) SD problems involve computing the (discounted) cumulative reward over time, see Eq. (3). However, DO problems have no expectation calculations. It only computes the best decision or variable values that yield the optimal objective value for a given scenario. However, the policy in SD problems aims to optimize the *expected* cost value.
5) The field of stochastic optimization, see [7], in general aims to determine a value for decision variables that works across all possible realizations of a system/problem on average. A stochastic optimization problem could optimize an expected objective function value or/and ensure constraints are satisfied on average with a certain probability. By contrast, SD problems aim to find a policy that performs optimally across different system realizations. Critically, an agent with such a policy changes its decisions as per system states.
6) As mentioned, DO problems are solved once for a given set of coefficients. By contrast, for SD problems, they can involve solving an optimization problem, e.g., a linear program, at each stage/time to determine the optimal reward for the current stage/time. Unlike DO problems, the goal is to determine the optimal decisions over sample paths. This means a SD problem may involve solving an optimization problem multiple times with the expected reward over sample paths used in Eq. (3).

4 Example Works

It is very important that the problem formulation of a paper spells out exactly and accurately the problem at hand. This gives the reader a sense of the final outcome and what a proposed solution solves. However, many works do not do what they preach. They formulate one problem type, and then solve it using a solution for a different problem type. As a result, these papers mislead readers. In fact, it is much better to skip their problem formulation and jump directly to their solution section.

Below, I list example works that formulate a deterministic optimization problem but then solve it using RL. The reader is encouraged to check the problem formulation of these works. Pay attention to the fact that they do not aim to compute or optimize a policy as they do not formulate a sequential decision problem! In fact, many of these problems as formulated can be solved via standard methods such as the Simplex algorithm. There is no need to employ RL. In addition, it is not clear whether their constraints should hold on average or with certain a probability, meaning they may have a constrained Markov decision process or safe reinforcement learning problem; otherwise, their constraints serve only to limit the action space of each state. It is important to understand how actions and system states affect constraints and vice-versa. Lastly, many works formulate a problem with no uncertainties, but then introduce uncertainties in their solution. This shows a lack of understanding of optimizing over stochastic processes.

To support the arguments above, please refer to the following flawed papers[1]: [12] [13] [14] [15] [16] [17] [18] [19] [20] [21] [22] [23] [24] [25] [26] [27] [28] [29] [30] [31] [32] [33] [2] [34] [35] [13] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49] [50] [51] [52] [53] [54] [55] [56] [57] [1] [58] [59] [60] [61] [62] [63] [64] [65] [66] [67] [68] [69] [3] [70] [71] [28] [72] [73] [74] [75] [76] [77] [78] [79] [80].

There are also works such as [81]–[90] that aim to minimize/maximize the average objective function value. However, these works do not specify how the said average is calculated as per the state and action taken by an agent. Moreover, it is unclear whether there are any time average requirements relating to constraints. Further, they do not state explicitly that their aim is to find a policy. Another issue can be seen in [91], namely it aims to find a policy but botched its problem formulation. The authors of [71], [92] first proposed a deterministic problem and then change their mind to address a sequential decision problem. Critically, the work in [71] misunderstood the definition of *observation* in the Partially Observable Markov Decision Process (POMDP). Many of these incorrect works have multiple agents, e.g., [68], [69], [77], [93], but yet they do not formulate a stochastic game; a good reference to learn stochastic games and multi-agent RL is [94].

In general, the cited works above show an incorrect or poor understanding of (i) basic mathematics, especially how expectation is calculated, (ii) basic mathematical programming formulations, (iii) what RL solves, and (iv) the definition of *policy*. In sum, these works do not do what they preach.

So what are examples of correct works? The reader is referred to [95]–[100] and [101]; notice that they have Eq. (3). There are also other ways to state a sequential decision problem accurately, see [102]–[104].

5 Conclusion

The motivation for this note is to highlight fundamental errors in many published works. It serves to educate future researchers that aim to use a reinforcement learning approach, and encourages them to have a better understanding of optimization theory and sequential decision processes. In this respect, the reader is referred to standard textbooks in both areas. The general advice is to learn optimization theory or/and reinforcement learning from reputable sources, not from research papers, including this document.

Feel free to recommend this document to authors and students, and especially when reviewing papers to apprise authors of the aforementioned critical issues.

References

[1] H. Peng and X. Shen, "Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2416–2428, Oct. 2020.

[2] J. Chen, P. Wan, and G. Xu, "Cooperative learning-based joint UAV and human courier scheduling for emergency medical delivery service," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 1, pp. 935–949, Jan. 2025.

[3] T. Du, X. Gui, and H. Dai, "An attention-driven heterogeneous multi-agent framework for uav and satellite-assisted task offloading in hybrid ground device networks," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

1. Only those in high impact journals are cited.

[4] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*. Wiley, Jan. 2013.

[5] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, Jul. 2006.

[6] J. Mattingley, Y. Wang, and S. Boyd, "Receding horizon control," *IEEE Control Systems*, vol. 31, no. 3, p. 52–65, Jun. 2011.

[7] J. R. Birge, *Introduction to Stochastic Programming*. Springer, Jun. 2011.

[8] W. B. Powell, *Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions*. Wiley, Apr. 2022.

[9] D. Bertsekas, *Reinforcement Learning and Optimal Control*. Athena Scientific, Jul. 2019.

[10] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, Apr. 1994.

[11] W. B. Powell and S. Meisel, "Tutorial on stochastic optimization in energy—part i: Modeling and policies," *IEEE Transactions on Power Systems*, vol. 31, no. 2, pp. 1459–1467, Mar. 2016.

[12] T. Liu, S. Ni, X. Li, Y. Zhu, L. Kong, and Y. Yang, "Deep reinforcement learning based approach for online service placement and computation resource allocation in edge computing," *IEEE Transactions on Mobile Computing*, vol. 22, no. 7, pp. 3870–3881, Jul. 2023.

[13] X. Wei, L. Cai, N. Wei, P. Zou, J. Zhang, and S. Subramaniam, "Joint UAV trajectory planning, DAG task scheduling, and service function deployment based on DRL in UAV-empowered edge computing," *IEEE Internet of Things Journal*, vol. 10, no. 14, pp. 12 826–12 838, Jul. 2023.

[14] Y. Wu, Z. Jia, Q. Wu, and Z. Lu, "Adaptive QoE-aware SFC orchestration in uav networks: A deep reinforcement learning approach," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 6, pp. 6052–6065, Dec. 2024.

[15] P. T. A. Quang, Y. Hadjadj-Aoul, and A. Outtagarts, "A deep reinforcement learning approach for VNF forwarding graph embedding," *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1318–1331, Dec. 2019.

[16] B. Li, R. Yang, L. Liu, and C. Wu, "Service placement and trajectory design for heterogeneous tasks in multi-UAV edge computing networks," *IEEE Internet Things J.*, pp. 1–1, 2024.

[17] J. Du, Z. Kong, A. Sun, J. Kang, D. Niyato, X. Chu, and F. R. Yu, "MADDPG-based joint service placement and task offloading in MEC empowered air–ground integrated networks," *IEEE Internet Things J.*, vol. 11, no. 6, pp. 10 600–10 615, 2024.

[18] Y. Li, J. Luo, Y. Ran, and J. Pi, "DeepISL: Joint optimization of leo inter-satellite link planning and power allocation via parameterized deep reinforcement learning," in *IEEE GLOBECOM*, Kuala Lumpur, Malaysia, Dec. 2023, pp. 3977–3982.

[19] L. Zhao, L. Li, Z. Tan, A. Hawbani, Q. He, and Z. Liu, "Multiagent deep-reinforcement-learning-based cooperative perception and computation in VEC," *IEEE Internet of Things Journal*, vol. 12, no. 12, pp. 21 350–21 363, Jun. 2025.

[20] D. Yu, X. Liu, J. Ning, S. Wang, C. Zhu, and W. Zhao, "Deep reinforcement learning-based ai task offloading in resource-constrained IIoT computing environments," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

[21] J. Hao, L. Wang, M. Odiathevar, W. K. G. Seah, G. Xu, B. Huang, and Y. Gao, "Prune-based deep reinforcement learning offloading algorithm for mobile edge computing," *IEEE Transactions on Cognitive Communications and Networking*, pp. 1–1, 2025.

[22] M. S. Munir, S. F. Abedin, N. H. Tran, and C. S. Hong, "When edge computing meets microgrid: A deep reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7360–7374, Oct. 2019.

[23] B. Xiao, C. Yu, X. Chen, Z. Chen, and G. Min, "Multi-agent collaboration for workflow task offloading in end-edge-cloud environments using deep reinforcement learning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 36, no. 11, pp. 2281–2296, Nov. 2025.

[24] X. Deng, H. Yang, J. Zhang, J. Gui, S. Lin, X. Wang, and G. Min, "Task offloading in internet of vehicles: A DRL-based approach with representation learning for DAG scheduling," *IEEE Transactions on Mobile Computing*, vol. 24, no. 6, pp. 5045–5060, 2025.

[25] X. Zhang, C. Wang, Y. Zhu, J. Cao, and T. Liu, "Multi-agent deep reinforcement learning with trajectory prediction for task migration-assisted computation offloading," *IEEE Transactions on Mobile Computing*, vol. 24, no. 7, pp. 5839–5856, Jul. 2025.

[26] Q. Liao, Z. Feng, H. Wu, S. Chen, and X. Xue, "Multi-objective self-organization scheduling of dynamic AAV-MD networks via two stage deep reinforcement learning," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 2, pp. 3826–3836, 2025.

[27] X. Li, X. Du, N. Zhao, and X. Wang, "Computing over the sky: Joint UAV trajectory and task offloading scheme based on optimization-embedding multi-agent deep reinforcement learning," *IEEE Transactions on Communications*, vol. 72, no. 3, pp. 1355–1369, Mar. 2024.

[28] S. Cheng, F. Feng, T. Bi, and T. Jiang, "Resource-aware reinforcement learning-based transmission optimization for mobile augmented reality in edge computing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 11, no. 4, pp. 2356–2367, Aug. 2025.

[29] H. Tong, C. Chen, W. Jiang, T. Wang, and J. Zhu, "Adaptive edge task offloading via parameterized multi-objective reinforcement learning with hybrid action space," *IEEE Transactions on Network Science and Engineering*, pp. 1–18, 2025.

[30] A. Lotfolahi and H.-W. Ferng, "DRL-based resource allocation in NOMA-aided industrial IoT towards energy productivity maximization," *IEEE Transactions on Network Science and Engineering*, pp. 1–16, 2025.

[31] H. Zhang, J. Du, C. Jiang, J. Wang, F. Bader, and M. Debbah, "Task offloading in UAV-assisted mobile cloud-edge computing networks: An AoP-aware HAPPO approach," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 9, pp. 14 745–14 759, 2025.

[32] P. Wan, G. Xu, J. Chen, and Y. Zhou, "Deep reinforcement learning enabled multi-UAV scheduling for disaster data collection with time-varying value," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 6691–6702, Jul. 2024.

[33] J. Fan, X. Chang, J. Mišić, V. B. Mišić, T. Yang, and Y. Gong, "Energy-constrained safe path planning for UAV-assisted data collection of mobile IoT devices," *IEEE Internet of Things Journal*, vol. 11, no. 24, pp. 39 971–39 983, Dec. 2024.

[34] H. Zeng, Z. Zhu, Y. Wang, Z. Xiang, and H. Gao, "Periodic collaboration and real-time dispatch using an actor–critic framework for UAV movement in mobile edge computing," *IEEE Internet of Things Journal*, vol. 11, no. 12, pp. 21 215–21 226, Jun. 2024.

[35] S. Malektaji, M. Rayani, A. Ebrahimzadeh, V. M. Raee, H. Elbiaze, and R. H. Glitho, "Dynamic joint VNF forwarding graph composition and embedding: A deep reinforcement learning framework," *IEEE Transactions on Network and Service Management*, vol. 20, no. 4, pp. 4615–4633, Dec. 2023.

[36] X. Yu, R. Wang, J. Hao, Q. Wu, C. Yi, P. Wang, and D. Niyato, "Priority-aware deployment of autoscaling service function chains based on deep reinforcement learning," *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 3, pp. 1050–1062, Jun. 2024.

[37] Y. Qin, Z. Zhang, X. Li, W. Huangfu, and H. Zhang, "Deep reinforcement learning based resource allocation and trajectory planning in integrated sensing and communications UAV network," *IEEE Trans. on Wirel. Commun.*, vol. 22, no. 11, pp. 8158–8169, Nov. 2023.

[38] X. Zhang, H. Tian, W. Ni, Z. Yang, and M. Sun, "Deep reinforcement learning for energy efficiency maximization in SWIPT-based over-the-air federated learning," *IEEE Transactions on Green Communications and Networking*, vol. 8, no. 1, pp. 525–541, Mar. 2024.

[39] Y. Liu, J. Yan, and X. Zhao, "Deep reinforcement learning based latency minimization for mobile edge computing with virtualization in maritime UAV communication network," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 4225–4236, Apr. 2022.

[40] C. Liu, Y. Zhong, R. Wu, S. Ren, S. Du, and B. Guo, "Deep reinforcement learning based 3D-trajectory design and task offloading in UAV-enabled MEC system," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 2, pp. 3185–3195, Feb. 2025.

[41] H. Zhang, Z. Tian, L. Zeng, L. Lu, S. Qiao, S. Chen, and X. Liu, "Distributed multiagent reinforcement learning approach for multiserver multiuser task offloading," *IEEE Internet of Things Journal*, vol. 12, no. 18, pp. 37 836–37 852, Sep. 2025.

[42] C. Pan, J. He, Z. Luo, K. Wang, Y. Yao, and X. Yue, "Federated deep reinforcement learning for delay and energy consumption tradeoff in scalable cell-free mobile edge computing networks," *IEEE Transactions on Green Communications and Networking*, pp. 1–1, 2025.

[43] F. Minani, M. Kobayashi, T. Fujihashi, M. A. Alim, S. Saruwatari, M. Nishi, and T. Watanabe, "Channel prediction and fair resource allocation for NTN uplinks by LSTM and deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 24, no. 10, pp. 8311–8330, Oct. 2025.

[44] H. Liu, W. Huang, D. I. Kim, S. Sun, Y. Zeng, and S. Feng, "Towards efficient task offloading with dependency guarantees in vehicular edge networks through distributed deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 9, pp. 13 665–13 681, Sep. 2024.

[45] C. Fang, C. H. Liu, H. Wang, G. Qi, Z. Liu, and D. Wu, "Multi-task-oriented emergency-aware UAV crowdsensing: A hierarchical multi-agent deep reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2025.

[46] X. Chen and G. Liu, "Energy-efficient task offloading and resource allocation via deep reinforcement learning for augmented reality in mobile edge networks," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10 843–10 856, Jul. 2021.

[47] S. Cheng, X. Lin, X. Li, and J. Wang, "Joint UAV trajectory and radcom task schedule for IVNs: A game-embedding multi-agent deep reinforcement learning approach," *IEEE Transactions on Wireless Communications*, vol. 24, no. 1, pp. 181–196, Jan. 2025.

[48] C. Li, K. Jiang, G. He, F. Bing, and Y. Luo, "A computation offloading method for multi-UAVs assisted MEC based on improved federated DDPG algorithm," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 12, pp. 14 062–14 071, Dec. 2024.

[49] X. Fu, X. Huang, Q. Pan, P. Pace, G. Aloi, and G. Fortino, "Cooperative data collection for UAV-assisted maritime IoT based on deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 7, pp. 10 333–10 349, Jul. 2024.

[50] R. Ding, J. Chen, W. Wu, J. Liu, F. Gao, and X. Shen, "Packet routing in dynamic multi-hop UAV relay network: A multi-agent learning approach," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 10 059–10 072, Sep. 2022.

[51] T. Du, X. Gui, X. Teng, K. Zhang, and D. Ren, "Dynamic trajectory design and bandwidth adjustment for energy-efficient UAV-assisted relaying with deep reinforcement learning in MEC IoT system," *IEEE Internet of Things Journal*, vol. 11, no. 23, pp. 37 463–37 479, Dec. 2024.

[52] M. M. Alam and S. Moh, "Joint trajectory control, frequency allocation, and routing for UAV swarm networks: A multi-agent deep reinforcement learning approach," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 11 989–12 005, Dec. 2024.

[53] X. Li, H. Liu, C. Li, G. Chen, C. Zhang, and Z. Y. Dong, "Deep reinforcement learning-based explainable pricing policy for virtual storage rental service," *IEEE Transactions on Smart Grid*, vol. 14, no. 6, pp. 4373–4384, 2023.

[54] C. Shih-Huan Hsu, J. Martín-Pérez, D. De Vleeschauwer, L. Valcarenghi, X. Li, and C. Papagianni, "A deep RL approach on task placement and scaling of edge resources for cellular vehicle-to-network service provisioning," *IEEE Transactions on Network and Service Management*, vol. 22, no. 4, pp. 3262–3280, Aug. 2025.

[55] S. Yao, M. Wang, J. Ren, T. Xia, W. Wang, K. Xu, M. Xu, and H. Zhang, "Multi-agent reinforcement learning for task offloading in crowd-edge computing," *IEEE Transactions on Mobile Computing*, vol. 24, no. 10, pp. 9289–9302, Oct. 2025.

[56] Z. Gong, O. Hashash, Y. Wang, Q. Cui, W. Ni, W. Saad, and K. Sakaguchi, "UAV-aided lifelong learning for AoI and energy optimization in nonstationary IoT networks," *IEEE Internet of Things Journal*, vol. 11, no. 24, pp. 39 206–39 224, Dec. 2024.

[57] X. Liu, H. Zhou, Z. Zhang, Q. Gao, and T. Ma, "Multipath cooperative routing in ultradense LEO satellite networks: A deep-reinforcement-learning-based approach," *IEEE Internet of Things Journal*, vol. 12, no. 2, pp. 1789–1804, Jan. 2025.

[58] Y. Ran, Y. Ding, S. Chen, J. Lei, and J. Luo, "Fully-distributed dynamic packet routing for LEO satellite networks: A GNN-enhanced multi-agent reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 3, pp. 5229–5234, Mar. 2025.

[59] Y. Lyu, H. Hu, R. Fan, Z. Liu, J. An, and S. Mao, "Dynamic routing for integrated satellite-terrestrial networks: A constrained multi-agent reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 5, pp. 1204–1218, May 2024.

[60] R. Mohamed, M. Avgeris, A. Leivadeas, and I. Lambadaris, "Optimizing resource fragmentation in virtual network function placement using deep reinforcement learning," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 2, pp. 1475–1491, Sep. 2024.

[61] Z. Xu, C. Luo, and R. Shao, "Zero-shot generalizable task offloading in edge computing: a causal reinforcement learning approach," *IEEE Transactions on Cloud Computing*, pp. 1–18, 2025.

[62] C. Zhong, M. C. Gursoy, and S. Velipasalar, "Deep multi-agent reinforcement learning based cooperative edge caching in wireless networks," in *IEEE ICC*, vol. May, Shanghai, China, 2019, pp. 1–6.

[63] X. Liao, J. Shi, Z. Li, L. Zhang, and B. Xia, "A model-driven deep reinforcement learning heuristic algorithm for resource allocation in ultra-dense cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 983–997, Jan. 2020.

[64] B. Zhao, G. Ren, X. Dong, and H. Zhang, "Distributed Q-learning based joint relay selection and access control scheme for IoT-oriented satellite terrestrial relay networks," *IEEE Communications Letters*, vol. 25, no. 6, pp. 1901–1905, Jun. 2021.

[65] J. Wang, L. Zhao, J. Liu, and N. Kato, "Smart resource allocation for mobile edge computing: A deep reinforcement learning approach," *IEEE Transactions on Emerging Topics in Computing*, vol. 9, no. 3, pp. 1529–1541, Sep. 2021.

[66] S. Liu, H. Yang, L. Xiao, M. Zheng, H. Lu, and Z. Xiong, "Learning-based resource management optimization for UAV-assisted MEC against jamming," *IEEE Transactions on Communications*, vol. 72, no. 8, pp. 4873–4886, Aug. 2024.

[67] K. Zhao, L. Peng, and B. Tak, "Joint DRL-based UAV trajectory planning and TEG-based task offloading," *IEEE Transactions on Consumer Electronics*, vol. 71, no. 2, pp. 3779–3789, May 2025.

[68] W. Jiang, Y. Zhan, and X. Fang, "Satellite edge computing for mobile multimedia communications: A multi-agent federated reinforcement learning approach," *ACM Trans. Auton. Adapt. Syst.*, Feb. 2025.

[69] M. Jia, L. Zhang, J. Wu, Q. Guo, G. Zhang, and X. Gu, "Deep multiagent reinforcement learning for task offloading and resource allocation in satellite edge computing," *IEEE Internet of Things Journal*, vol. 12, no. 4, pp. 3832–3845, Feb. 2025.

[70] M. Akbari, A. Syed, W. S. Kennedy, and M. Erol-Kantarci, "Constrained federated learning for AoI-limited SFC in UAV-aided MEC for smart agriculture," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 1, pp. 277–295, 2023.

[71] H. Huang, J. Liang, and G. Min, "Joint DNN model deployment, selection, and configuration for heterogeneous inference services toward edge intelligence," *IEEE Transactions on Mobile Computing*, vol. 24, no. 11, pp. 12 726–12 741, Nov. 2025.

[72] A. Xu, Z. Hu, X. Li, B. Chen, H. Xiao, X. Zhang, H. Zheng, X. Feng, M. Zheng, P. Zhong, and K. Li, "CoMS: Collaborative DNN model selection for heterogeneous edge computing systems," *IEEE Transactions on Vehicular Technology*, vol. 74, no. 2, pp. 3172–3184, Feb. 2025.

[73] Z. Wang, M. Goudarzi, and R. Buyya, "TF-DDRL: A transformer-enhanced distributed DRL technique for scheduling IoT applications in edge and cloud computing environments," *IEEE Transactions on Services Computing*, vol. 18, no. 2, pp. 1039–1053, Apr. 2025.

[74] W. Tan, T. Ding, and L. Liu, "Intelligent UAV deployment for energy-efficient IoT data collection," *IEEE Internet of Things Journal*, vol. 12, no. 17, pp. 34 890–34 899, Sep. 2025.

[75] W. Zhang, D. Yang, H. Peng, W. Wu, W. Quan, H. Zhang, and X. Shen, "Deep reinforcement learning based resource management for DNN inference in industrial IoT," *IEEE Trans. on Vehic. Techn.*, vol. 70, no. 8, pp. 7605–7618, Aug. 2021.

[76] X. Song, J. Feng, L. Liu, Q. Pei, F. Richard Yu, and N. Zhang, "A deep reinforcement learning with transformer integration for directed acyclic graph scheduling in edge networks," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2025.

[77] Y. Zhu, H. Yao, T. Mai, W. He, N. Zhang, and M. Guizani, "Multiagent reinforcement-learning-aided service function chain deployment for internet of things," *IEEE Internet of Things Journal*, vol. 9, no. 17, pp. 15 674–15 684, Sep. 2022.

[78] L. Wang, X. Liu, H. Ding, Y. Hu, K. Peng, and M. Hu, "Energy-delay-aware joint microservice deployment and request routing with DVFS in edge: A reinforcement learning approach," *IEEE Transactions on Computers*, vol. 74, no. 5, pp. 1589–1604, May 2025.

[79] Q. Wang, X. Tong, Y. Li, C. Wang, and C. Zhang, "Integrated scheduling optimization for automated container terminal: A reinforcement learning-based approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 7, pp. 10 019–10 035, Jul. 2025.

[80] A. Islam, M. Ghose, and S. Pasricha, "An RL-based framework for task offloading and resource allocation in energy harvesting-based multi-access edge computing," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2025.

[81] H. Li, Y. Wang, M. Pan, S. Li, and W. Guan, "Deep reinforcement learning based joint resource allocation and service migration for smart-buoy-enabled maritime multi-access edge computing networks," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

[82] Y. Li, X. Zhao, and H. Liang, "Throughput maximization by deep reinforcement learning with energy cooperation for renewable ultradense IoT networks," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9091–9102, Sep. 2020.

[83] C. Qiu, Y. Hu, Y. Chen, and B. Zeng, "Deep deterministic policy gradient (ddpg)-based energy harvesting wireless communications," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8577–8588, Oct. 2019.

[84] M. K. Sharma, A. Zappone, M. Assaad, M. Debbah, and S. Vassilaras, "Distributed power control for large energy harvesting networks: A multi-agent deep reinforcement learning approach," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 4, pp. 1140–1154, Dec. 2019.

[85] Q. V. Do, Q.-V. Pham, and W.-J. Hwang, "Deep reinforcement learning for energy-efficient federated learning in UAV-enabled wireless powered networks," *IEEE Communications Letters*, vol. 26, no. 1, pp. 99–103, Jan. 2022.

[86] S. Zhu, B. Zhu, K. Chi, K. Yu, and S. Mumtaz, "Long-term computation rate maximization in UAV-enabled wirelessly powered MEC," *IEEE Transactions on Communications*, pp. 1–1, 2025.

[87] J. Wan, S. Lin, Z. Zhang, J. Zhang, and T. Zhang, "Scheduling real-time wireless traffic: A network-aided offline reinforcement learning approach," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 22 331–22 340, Dec. 2023.

[88] K. Gao, J. Du, C. Jiang, J. Simonjan, D. Mishra, C. Zhang, and M. Debbah, "Cooperative DNN partitioning in energy-harvesting and MEC-enabled AAV networks," *IEEE Internet of Things Journal*, vol. 12, no. 13, pp. 24 329–24 344, Jul. 2025.

[89] Y. Zhang, W. Zhang, M. Yuan, L. Xu, C. Yan, T. Gong, and H. Du, "Lightweight configuration adaptation with multi-teacher reinforcement learning for live video analytics," *IEEE Transactions on Mobile Computing*, vol. 24, no. 5, pp. 4466–4480, May 2025.

[90] N. Cheng, H. Chen, R. Sun, L. Ma, C. Zhou, Y. Zhang, and Y. Hui, "Value-of-information optimization for object detection-driven joint video transmission and processing in UAV-enabled wireless networks," *IEEE Journal on Miniaturization for Air and Space Systems*, vol. 6, no. 2, pp. 59–69, Jun. 2025.

[91] M. Ejaz, J. Gui, M. Asim, M. A. El-Affendi, C. Fung, and A. A. Abd El-Latif, "RL-planner: Reinforcement learning-enabled efficient path planning in multi-uav MEC systems," *IEEE Transactions on Network and Service Management*, vol. 21, no. 3, pp. 3317–3329, Jun. 2024.

[92] X. Jing, R. Wang, H. Lei, H. Liu, and Q. Chen, "Multi-agent discrete soft actor-critic algorithm-based multi-user collaborative anti-jamming strategy," *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 5025–5038, 2025.

[93] Q. Jiang, P. Han, X. Xin, and K. Chen, "Deep reinforcement learning and edge computing for multisatellite on-orbit task scheduling," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 61, no. 5, pp. 14 143–14 159, Oct. 2025.

[94] D. Bertsekas, "Multiagent reinforcement learning: Rollout and policy iteration," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 249–272, Feb. 2021.

[95] I. Rahmaty, H. Shah-Mansouri, and A. Movaghar, "QECO: a QoE-oriented computation offloading algorithm based on deep reinforcement learning for mobile edge computing," *IEEE Transactions on Network Science and Engineering*, vol. 12, no. 4, pp. 3118–3130, Jul. 2025.

[96] J. Li, Q. Jiang, V. C. M. Leung, Z. Ma, and K. Kwarteng Abrokwa, "Deep-reinforcement-learning-based joint optimization of task migration and resource allocation for mobile-edge computing," *IEEE Internet of Things Journal*, vol. 12, no. 13, pp. 24 431–24 440, Dec. 2025.

[97] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, "Deep reinforcement learning for dynamic multichannel access in wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 257–265, Jun. 2018.

[98] C. Jiang and X. Zhu, "Reinforcement learning based capacity management in multi-layer satellite networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4685–4699, Jul. 2020.
[99] N. Yang, J. Wen, M. Zhang, and M. Tang, "Generalizable pareto-optimal offloading with reinforcement learning in mobile edge computing," *IEEE Transactions on Services Computing*, pp. 1–13, 2025.
[100] Y. Luo and K.-W. Chin, "An energy efficient channel bonding and transmit power control approach for WiFi networks," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8251–8263, Aug. 2021.
[101] M. Tang and V. W. Wong, "Deep reinforcement learning for task offloading in mobile edge computing systems," *IEEE Transactions on Mobile Computing*, vol. 21, no. 6, pp. 1985–1997, Jun. 2022.
[102] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 6, pp. 1133–1146, Jun. 2020.
[103] H. Yang and X. Xie, "An actor-critic deep reinforcement learning approach for transmission scheduling in cognitive Internet of things systems," *IEEE Systems Journal*, vol. 14, no. 1, pp. 51–60, Mar. 2020.
[104] Y. Liu, Y. Lu, X. Li, W. Qiao, Z. Li, and D. Zhao, "SFC embedding meets machine learning: Deep reinforcement learning approaches," *IEEE Communications Letters*, vol. 25, no. 6, pp. 1926–1930, Jun. 2021.