



Maximizing SAGIN coverage with joint HAP/UAV placement, power, bandwidth, and beamforming via deep actor-critic with a temperature function-based policy

Vitou That¹ · Sengly Muy² · Jung-Ryun Lee^{2,3}

Received: 7 November 2025 / Accepted: 13 January 2026 / Published online: 24 February 2026
© The Author(s) 2026

Abstract

We address a coverage optimization problem in a space-air-ground integrated network (SAGIN) with high-altitude platforms (HAPs) and unmanned aerial vehicles (UAVs), where the joint optimization of aerial node placement, transmit power control, bandwidth allocation, and beamforming results in a high-dimensional and strongly coupled optimization problem. To address this challenge, we propose a deep actor-critic (DAC)-based framework capable of handling joint continuous control in complex SAGIN environments and adapting to dynamic network conditions. To accelerate training convergence, we incorporate a temperature function that identifies the most frequently used actions and enhances the exploration process of the DAC algorithm. The performance of the proposed algorithm is evaluated through simulations conducted in a realistic network scenario on Jeju Island, South Korea. We compare the performance of the proposed algorithm against benchmarks, including a soft actor-critic (SAC), deep deterministic policy gradient (DDPG), deep actor-critic (DAC), genetic algorithm (GA), and a gradient search (GS). The results demonstrate that our proposed algorithm achieves higher coverage performance and provides more efficient bandwidth utilization, faster convergence, and a slightly improved data rate.

Keywords HAP placement · Temperature function · Beamforming · SAGIN coverage

1 Introduction

Satellite communication has recently gained attention from researchers as an effective way to improve global coverage, driven by the successful deployment of numerous low Earth orbit (LEO) satellites by Starlink [14]. To address the growing demand for high-density coverage areas in satel-

lite communication, technological advances such as multiple beamforming, frequency reuse, and power control have been studied. These technologies enhance satellite communication by providing reliable, high-speed connectivity for ground users (GUs) [9]. Despite recent advances, there remain challenges to achieve seamless coverage, particularly in remote or urban areas where user locations can affect signal quality. To tackle this challenge and improve the performance of satellite communication systems, space-air-ground integrated network (SAGIN) has been proposed, which integrates satellites with aerial-based technologies, such as high-altitude platforms (HAPs) and unmanned aerial vehicles (UAVs) [4]. HAPs and UAVs in SAGIN offer dynamic and flexible coverage, which can address coverage gaps that satellites or terrestrial networks alone might miss. This adaptability is particularly critical for providing seamless connectivity in areas with varying user densities and fluctuating demands. However, this flexibility introduces additional challenges, particularly in managing network resources.

In SAGIN, bandwidth allocation becomes more complex compared to traditional satellite-based systems, as it must dynamically adapt to the user density of the coverage

✉ Jung-Ryun Lee
jrlee@cau.ac.kr

Vitou That
vitou1707@cau.ac.kr

Sengly Muy
maysengly@cau.ac.kr

¹ Department of Intelligent Energy Industry Convergence at Chung-Ang University, 84 Heuksuk-ro, Dongjak-gu Seoul 17546, Republic of Korea

² Department of Intelligent Energy and Industry at Chung-Ang University, Dongjak-gu, Seoul 17546, Republic of Korea

³ School of Electrical and Electronics Engineering at Chung-Ang University, Dongjak-gu, Seoul 17546, Republic of Korea

area. HAPs and UAVs, as reconfigurable aerial platforms, are required to be properly positioned and to precisely control their beamforming centers to address coverage gaps and respond to high-density areas, while maintaining reliable connectivity with the satellite layer. In addition, UAVs face power constraints that require careful power control strategies to balance their limited energy resources with the need to maintain adequate coverage, while bandwidth allocation further challenges minimum-rate satisfaction under dense GU deployments [17]. These challenges require advanced adaptive approaches to effectively integrate HAPs and UAVs into SAGIN while ensuring reliable and efficient network performance. To effectively handle this complex problem, algorithms that take advantage of artificial intelligence (AI) and machine learning (ML) become essential, which can play a critical role in coverage optimization while ensuring adaptive coordination among these diverse components.

On the other hand, Deep Reinforcement Learning (DRL) has emerged as a powerful paradigm for solving complex optimization problems in dynamic environments by combining the decision-making capabilities of reinforcement learning with the function approximation power of deep neural networks. DRL has demonstrated remarkable success in various wireless communication applications, including resource allocation and network management, due to its ability to adapt to changing conditions without prior system knowledge [11]. However, a critical challenge in DRL is the exploration-exploitation trade-off, where traditional exploration strategies like ϵ -greedy rely on uniform random sampling, which can be inefficient in high-dimensional action spaces and often lead to premature convergence to local optima, particularly in complex multi-objective scenarios [36].

Many studies have addressed the significant issues of SAGIN to enhance its communication capabilities. In [12], the authors proposed the multi-attribute decision-making (MADM) algorithm with graph-based handover (GBH) to improve SAGIN's overall throughput and access failure probability. The authors of [23] have studied the performance of SAGIN considering a UAV-assisted satellite communication with an asymmetric dual-hop radio frequency/FSO system to minimize the average bit error under high user-device density. In [7], the authors aimed to maximize the sum-rate problem of SAGIN by employing successive convex approximations and semi-definite programming. The work in [5] proposed a multi-objective genetic algorithm to enhance the coverage of the target areas in SAGIN while simultaneously minimizing the total number of satellites in the constellation. UAV deployment problems have also been extensively studied, including recent cooperative UAV-assisted systems with joint three-dimensional deployment and resource optimization using heuristic or evolutionary algorithms [21]. However, these existing approaches have several limitations. Most works typically optimize isolated

control parameters rather than joint resource allocation, and rely on traditional optimization methods that may struggle with the high-dimensional, non-convex nature of SAGIN optimization problems.

In addition, many studies have leveraged ML techniques to enhance SAGIN performance through optimized resource allocation. In [32], the authors studied the end-to-end anomaly traffic detection algorithm for SAGIN by proposing a semi-supervised ML method. In [34], the authors proposed an RL-assisted bandwidth-aware virtual network resource allocation algorithm (RL-BA-VNA), which transforms the network resource allocation problem in SAGIN into a multi-domain virtual network resource allocation problem to optimize the data rate of the system. The authors of [27] proposed a double Q-learning algorithm with better delay-sensitive replay memory (DSRPM) for offloading traffic for SAGIN, with the objective of reducing the time needed to collect data from the moving nodes around the network.

In [6], the authors proposed a network selection algorithm based on a deep deterministic policy gradient (DDPG) to improve resource utilization and achieve load balance in SAGIN. While these ML-based approaches show promising results, they suffer from exploration inefficiency due to reliance on conventional exploration strategies like ϵ -greedy, which can lead to premature convergence in complex multi-objective scenarios. Moreover, most existing works do not jointly optimize the critical factors of UAV positioning, power allocation, beamforming, and bandwidth management in an integrated framework.

In this work, we aim to improve SAGIN coverage through a comprehensive optimization framework that addresses multiple critical factors simultaneously. A Deep Actor-Critic (DAC) algorithm combined with a temperature function is employed to solve the problem. The proposed approach leverages a temperature-guided exploration mechanism inspired by simulated annealing and generalized annealing theory [30]. Its purpose is to regulate exploration diversity in high-dimensional continuous action spaces and reduce premature convergence, leading to faster and more stable training and improved final performance.

The idea and the methodology in our study can be summarized as follows.

1. We consider a SAGIN coverage optimization problem over a realistic service region, where multiple coupled continuous control variables, including aerial deployment, transmit power, beamforming direction, and bandwidth allocation, are jointly optimized to satisfy quality of service (QoS) requirements of GUs.
2. To address this complex optimization problem, we propose a temperature-guided deep actor-critic (DAC-Temp) algorithm, in which a temperature function is incorporated into the exploration process to regulate exploration

diversity and exploration step size, thereby improving learning behavior in high-dimensional continuous action spaces.

3. To evaluate the performance of our approach, we conduct extensive simulations and compare it against both traditional and learning-based baselines. Specifically, we implement two heuristic methods, Genetic Algorithm (GA) and Gradient Search (GS), as optimization-based references. We also benchmark our method against multiple DRL algorithms, including traditional DAC, Soft Actor-Critic (SAC) and DDPG. Simulation results show that the proposed DAC-Temp method consistently achieves higher coverage and data rate efficiency compared to all baseline algorithms while maintaining stable convergence. To the best of our knowledge, existing studies have not investigated SAGIN coverage optimization under this integrated formulation using a temperature-guided DAC framework.

The remainder of the paper is structured as follows. Section 2 presents the system model and problem formulation, including key optimization variables and constraints relevant to the SAGIN coverage problem. Section 3 introduces the proposed solution methodology, detailing the DAC-Temp algorithm and its temperature-based exploration strategy.

The section also provides the implementation details of the baseline methods including GA, GS, DAC, SAC, and DDPG. Section 4 offers a comprehensive performance evaluation through simulations conducted on Jeju Island, South Korea, comparing DAC-Temp with other approaches in terms of coverage, convergence, and data rate. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2 System Model and Problem Formulation

In this study, we consider a SAGIN network scenario that consists of a LEO satellite, H HAPs, U UAVs, and G GUs. The LEO satellite, HAPs, and UAVs are equipped with array antennas with C beamformings to improve the coverage of the SAGIN. Downlink communication from the LEO satellite to the GUs can be categorized by four types of connections: LEO-GU link, LEO-HAP-GU link, LEO-UAV-GU link, and LEO-HAP-UAV-GU link, as illustrated in Fig. 1. Link establishment is abstracted at the time-slot level, where each GU associates with the available downlink path that maximizes the achievable data rate subject to the minimum rate constraint (Table 1). Table 1 summarizes the mathematical notations and symbols used in this work.

The system operates in discrete time slot indexed by t . Let the indices of HAP, UAV, and GU be $h \in \mathcal{H} =$

Fig. 1 SAGIN system model: a LEO satellite cooperates with H HAPs and U UAVs, each using C -beam antennas, to serve G ground users in a region. Connections can be direct or multi-hop via HAPs/UAVs (LEO-GU, LEO-HAP-GU, LEO-UAV-GU, LEO-HAP-UAV-GU)

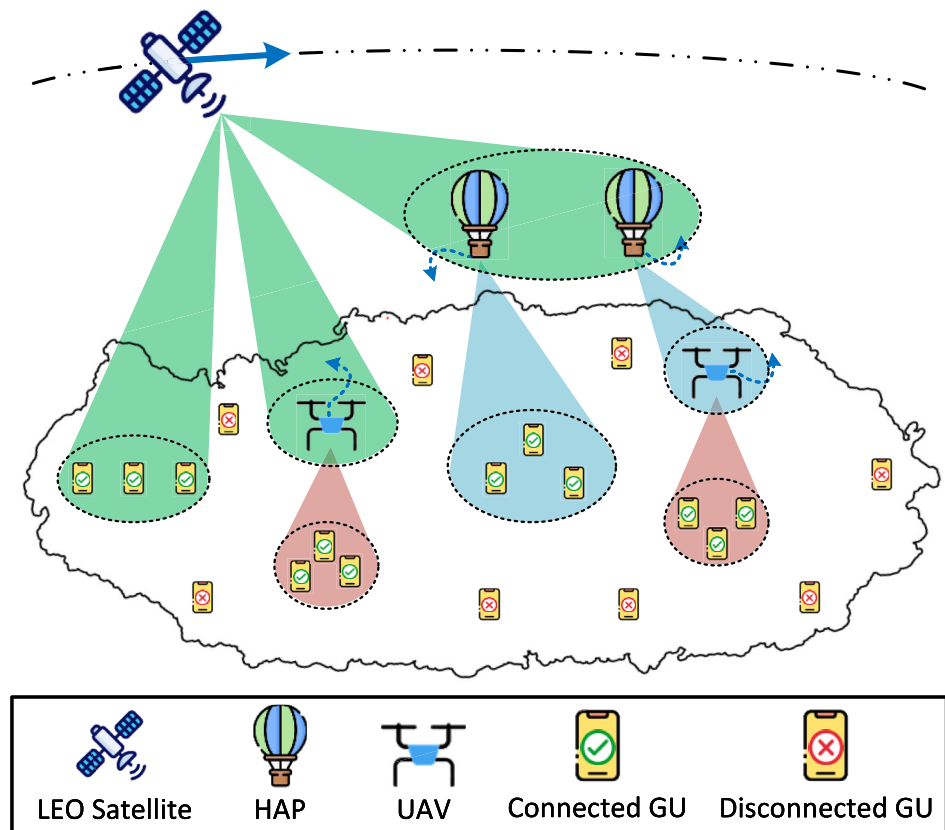


Table 1 List of symbols and variables

Symbols	Description
H, U, G, C	Number of HAPs, UAVs, GUs, and beamforming centers.
$\mathcal{H}, \mathcal{U}, \mathcal{G}, \mathcal{C}$	Index sets of HAPs, UAVs, GUs, and beamforming centers.
t	Time slot index.
$L_{LEO}(t), L_h(t), L_u(t), L_g$	Location of the LEO satellite, HAP h , UAV u , and GU g .
$Z_{LEO}, Z_{HAP}, Z_{UAV}$	Altitudes of LEO satellite, HAPs, and UAVs.
$L_{LEO,c}(t)$	Beamforming center of LEO satellite.
$L_{h,c}(t), L_{u,c}(t)$	Beamforming centers of HAPs and UAVs.
Ω	Geographical service region.
$G_{TX}(t), G_{RX}$	Transmit and receive antenna gains.
$PL_{TX,RX}(t), FD_{TX,RX}(t)$	Path loss and fading between tx and rx .
$d_{tx,rx}(t)$	Distance between transmitter and receiver.
$p_{LEO}(t), p_h(t), p_u(t)$	Transmit power of LEO, HAPs, and UAVs.
$p_{\max}^{LEO}, p_{\max}^{HAP}, p_{\max}^{UAV}$	Maximum transmit power limits.
$b_{LEO}(t), b_h(t), b_u(t)$	Bandwidth allocation ratios.
B	Total system bandwidth.
$\Gamma_{tx,rx}(t)$	SINR from transmitter tx to receiver rx .
$R_{tx,rx}(t), R_g(t)$	Link data rate and end-to-end GU data rate. rx .
$R_{\min}$	Minimum required data rate.
$a_g(t)$	Connectivity indicator of GU g .
$CON(t)$	Number of connected GUs at time t .
$L(t), B(t), P(t)$	Location, bandwidth, and power control variables.

$\{1, 2, \dots, H\}$, $u \in \mathcal{U} = \{1, 2, \dots, U\}$, and $g \in \mathcal{G} = \{1, 2, \dots, G\}$, respectively. The locations of the LEO satellite, HAP h , UAV u , and GU g in Cartesian coordinates are denoted as $L_{LEO}(t) = (x(t), y(t), Z_{LEO})_{LEO}$, $L_h(t) = (x(t), y(t), Z_{HAP})_h$, $L_u(t) = (x(t), y(t), Z_{UAV})_u$, and $L_g = (x, y, 0)_g$, respectively. Here, Z_{LEO} , Z_{HAP} , and Z_{UAV} represent the altitudes of the LEO satellite, HAPs, and UAVs, respectively. To simplify the network scenario, we assume that the altitudes of the LEO satellite, the HAPs and UAVs are fixed. Moreover, the LEO satellite is assumed to move along a predefined orbital trajectory with uniform motion, where its horizontal position varies over time at a constant orbital speed V_s . Let the location of the beamforming center on the ground be denoted as $L_{LEO,c}(t) = (x(t), y(t), 0)_{LEO,c}$, $L_{h,c}(t) = (x(t), y(t), 0)_{h,c}$, and $L_{u,c}(t) = (x(t), y(t), 0)_{u,c}$, respectively, where $c \in \mathcal{C} = \{1, 2, \dots, C\}$. It is noted that the coordinates x and y for the locations of HAPs, UAVs, and GUs are confined within a polygon Ω , which represents a geographically bounded region used for tractable regional modeling of a province or city [20].

2.1 Channel model

In this work, a single channel expression is adopted for all downlink hops, while differences between LEO satel-

lite, HAP, and UAV links are captured through link-specific parameters such as altitude, propagation distance, path-loss exponent, fading characteristics, and antenna configurations. All transmitter devices use spot beams generated by array antennas to transmit data to receivers. The LEO satellite acts as the transmitter tx , while GUs serve as the receiver rx . HAPs and UAVs function as transceivers, operating as either transmitters (tx) or receivers (rx). The channel model used in this scenario follows the transmitter's antenna gain $G_{TX}(t)$, path loss $PL_{TX,RX}(t)$, fading $FD_{TX,RX}(t)$, and the receiver's antenna gain G_{RX} . Based on this, the channel model from transmitter tx to receiver rx is expressed as

$$G_{tx,rx}(t) = G_{TX}(t) \times PL_{tx,rx}(t) \times FD_{tx,rx}(t) \times G_{RX}. \quad (1)$$

Equation 1 represents a unified channel model applicable to all links. Here, the transmitter tx and receiver rx vary depending on the considered hop, including LEO-HAP, LEO-UAV, LEO-GU, HAP-UAV, HAP-GU, and UAV-GU links.

From [13], the antenna gain from transmitter tx to receiver rx can be approximated as

$$G_{tx}(t) \approx G_{\max} - 12 \left(\frac{\angle(\vec{v}_{tx,c}(t), \vec{v}_{tx,rx}(t))}{\theta_{tx,3dB}} \right)^2, \quad (2)$$

where $G_{\max}$ is the maximum antenna gain observed at the beam center and $\theta_{tx,3dB}$ is the half-power beam-width. Here,

the angle between the direction from transmitter tx to beamforming center c and from transmitter tx to receiver rx is calculated as

$$\angle(\vec{v}_{tx,c}(t), \vec{v}_{tx,rx}(t)) = \cos^{-1} \left(\frac{\vec{v}_{tx,c}(t) \cdot \vec{v}_{tx,rx}(t)}{\|\vec{v}_{tx,c}(t)\| \|\vec{v}_{tx,rx}(t)\|} \right), \quad (3)$$

where $\vec{v}_{tx,c}(t) = (L_{tx,c}(t) - L_{tx}(t))$ and $\vec{v}_{tx,rx}(t) = (L_{rx}(t) - L_{tx}(t))$ are vectors from the transmitter tx to the beamforming center c and from the transmitter tx to the receiver rx , respectively. For simplicity, we assume all GUs share the same receiver antenna gain G_{RX} [26].

The path loss from transmitter tx to receiver rx can be defined as

$$PL_{tx,rx}(t) = \frac{\beta_0}{(d_{tx,rx}(t))^\alpha}, \quad (4)$$

where α is the path-loss exponent, β_0 is the channel power at a reference distance of 1 meter, and $d_{tx,rx}(t)$ is the distance between the transmitter tx and receiver rx . The distance between the transmitter tx and receiver rx is denoted as $d_{tx,rx}(t) = \|L_{rx}(t) - L_{tx}(t)\|$, where $\|\cdot\|$ denotes the Euclidean norm. $FD_{tx,rx}(t)$ represents the channel fading from transmitter tx to receiver rx , which is characterized by small-scale fading that follows a Rician distribution due to the predominant line-of-sight (LoS) propagation from elevated platforms to GUs in SAGIN downlink scenarios [31].

2.2 Data rate

Let $p_{tx}(t)$ be the transmit power of the transmitter tx at time t , where $tx \in \{\text{LEO}, \text{HAP}, \text{UAV}\}$. Accordingly, $p_{tx}(t)$ denotes $p_{LEO}(t)$, $p_h(t)$, or $p_u(t)$ depending on the transmitting node of the considered link. The received signal-to-interference-plus-noise ratio (SINR) from the transmitter tx to the receiver rx is expressed as

$$\Gamma_{tx,rx}(t) = \frac{p_{tx}(t) \times G_{tx,rx}(t)}{I_{tx'}(t) + \sigma_A^2}, \quad (5)$$

where $I_{tx'}(t)$ is the interference from other transmitters tx' , and σ_A^2 is the power spectral density of additive white Gaussian noise (AWGN). By using the Shannon–Hartley theorem, the data rate from the transmitter tx to the receiver rx is calculated as

$$R_{tx,rx}(t) = b_{tx}(t) B \log_2(1 + \Gamma_{tx,rx}(t)), \quad (6)$$

where $b_{tx}(t)$ is the bandwidth allocation of the transmitter tx , and B is the total bandwidth.

2.3 End-to-end (E2E) data rate

The data rates of the downlink communication connections at time t are explicitly defined as follows. The direct LEO-GU link data rate is given by $R_{LEO,g}(t)$. For relay-assisted links, the end-to-end data rates are determined by the bottleneck hop, including $R_{LEO,h,g}(t) = \min\{R_{LEO,h}(t), R_{h,g}(t)\}$, $R_{LEO,u,g}(t) = \min\{R_{LEO,u}(t), R_{u,g}(t)\}$, and $R_{LEO,h,u,g}(t) = \min\{R_{LEO,h}(t), R_{h,u}(t), R_{u,g}(t)\}$. The minimum operation reflects the bottleneck constraint in multi-hop communications, where the end-to-end data rate is limited by the weakest link in the transmission chain. Assuming that the GU g selects the connection with the highest data rate among all possible links, the end-to-end (E2E) data rate for GU g is expressed as

$$R_g(t) = \max_{\substack{h \in \mathcal{H}, \\ u \in \mathcal{U}}} \left\{ \begin{array}{ll} R_{LEO,g}(t), & R_{LEO,h,g}(t), \\ R_{LEO,u,g}(t), & R_{LEO,h,u,g}(t) \end{array} \right\}. \quad (7)$$

2.4 Coverage model

This study aims to improve SAGIN coverage by increasing the number of GUs that satisfy the minimum data rate requirement. Let $R_{\min}$ represent the minimum data rate required for GU satisfaction. GU g is assumed to be connected to the satellite when the data rate of GU g is greater than or equal to $R_{\min}$. Then, the indicator for the connection from the LEO satellite to GU g at time t is defined as

$$a_g(t) = \begin{cases} 1, & R_g(t) \geq R_{\min} \\ 0, & \text{otherwise} \end{cases}. \quad (8)$$

Then, the total number of GUs connected to the satellite at time t is given by

$$\text{CON}(t) = \sum_{g \in \mathcal{G}} a_g(t). \quad (9)$$

2.5 Problem formulation

The total number of GUs satisfying the minimum data rate requirement is influenced by several factors, such as the locations of the HAPs and UAVs ($L_h(t)$, $L_u(t)$), and the beamforming center locations for LEO, HAPs, and UAVs ($L_{LEO,c}(t)$, $L_{h,c}(t)$, $L_{u,c}(t)$). Furthermore, it depends on the bandwidth allocation for LEO, HAPs, UAVs ($b_{LEO}(t)$, $b_h(t)$, $b_u(t)$), as well as their respective transmit power ($p_{LEO}(t)$, $p_h(t)$, $p_u(t)$).

To simplify the notation, we define the set of control variables for locations as $\mathbf{L}(t) = \{L_h(t), L_u(t), L_{LEO,c}(t), L_{h,c}(t), L_{u,c}(t)\}$, the set of control variables for bandwidth

allocation as $\mathbf{B}(t) = \{b_{LEO}(t), b_h(t), b_u(t)\}$, and the set of control variables for transmit power as $\mathbf{P}(t) = \{p_{LEO}(t), p_h(t), p_u(t)\}$. Using the above notation, we can formulate the coverage problem in SAGIN as

$$(P1) : \max_{L(t), \mathbf{B}(t), \mathbf{P}(t)} \text{CON}(t) \quad (10)$$

s.t.

$$(C1) : L_h(t), L_u(t), L_{LEO,c}(t), L_{h,c}(t), L_{u,c}(t) \in \Omega$$

$$(C2) : b_{LEO}(t), b_h(t), b_u(t) \in [0, 1],$$

$$(C3) : p_{LEO}(t) \in [0, P_{\max}^{LEO}], p_h(t) \in [0, P_{\max}^{HAP}],$$

$$p_u(t) \in [0, P_{\max}^{UAV}],$$

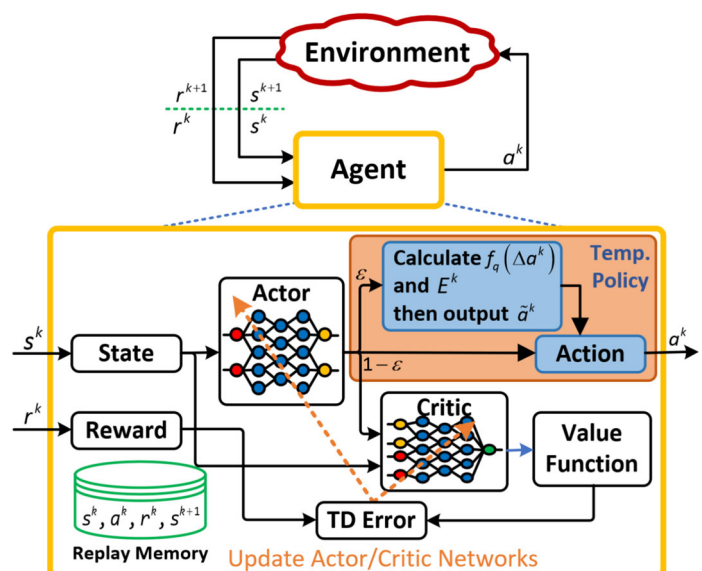
$$(C4) : R_{LEO,h}(t) \geq \sum_{g \in \mathcal{G}} R_{h,g}(t),$$

$$R_{LEO,u}(t) \geq \sum_{g \in \mathcal{G}} R_{u,g}(t),$$

$$R_{h,u}(t) \geq \sum_{g \in \mathcal{G}} R_{u,g}(t),$$

where $P_{\max}^{LEO}$, $P_{\max}^{HAP}$, and $P_{\max}^{UAV}$ are the maximum transmit power of the LEO satellite, HAPs, and UAVs, respectively. Constraint C1 represents geographical restrictions by requiring that the positions of HAPs, UAVs, GUs, and all beam-forming centers lie within the defined service area. Constraint C2 ensures that the allocated bandwidth for all transmitters, including LEO satellite, HAPs, and UAVs, lies within normalized bounds $[0, 1]$. Constraint C3 imposes power constraints by limiting the transmit power not to exceed their respective maximum power capacities. Finally, constraint C4 represents a backhaul capacity constraint under a relay-forwarding assumption, ensuring that the upper-layer link capacity is no smaller than the aggregate access rate served by the relay. Consequently, the end-to-end transmission rate is determined by the bottleneck among the involved hops.

Fig. 2 Architecture of the proposed DAC-Temp algorithm. The agent observes the current state $s(k)$, selects an action $a(k)$ via a temperature function-based policy that explores by accepting ϵ -guided trial actions using $f_q(\Delta a_k)$ and exploits by choosing the argmax-Q action with probability $1 - \epsilon$, interacts with the environment to receive reward $r(k)$ and next state $s(k+1)$, stores transitions in replay memory, and updates the actor and critic networks using temporal-difference (TD) error and policy gradient with SGD



3 Proposed optimization approaches

To address the problem in (P1), we propose a DRL approach that can learn and adapt to the complexities of dynamic environments [10]. The environment involves multiple tightly coupled continuous control variables, resulting in a high-dimensional action space that poses challenges for conventional optimization and exploration strategies. Specifically, we employ the DAC algorithm as illustrated in Fig. 2. Actor-critic learning is adopted to handle continuous-valued control decisions, where the critic network provides value-based guidance to stabilize policy learning under the non-convex optimization problem. By utilizing a Divide-and-Conquer strategy in DAC, complex optimization problems can be decomposed into smaller, independent subproblems. This not only reduces computational overhead but also enables efficient and parallelizable solutions, which makes it particularly effective in dynamic and heterogeneous network environments [18, 33].

However, the DAC algorithm in our work involves many control variables, which results in long training times to obtain the solution. To improve training efficiency, we integrate a temperature function into the DAC algorithm whose role is to explicitly control exploration diversity during learning, particularly in high-dimensional continuous action spaces where conventional random exploration becomes inefficient. The temperature function proposed in this work detects repeated actions, thus improving the exploration process and overall performance. This modification enables more efficient and accurate decision-making in complex environments, together with the DAC algorithm.

3.1 Designing of the DRL model

The DRL model consists of three main components: state, action, and reward. In this formulation, the state consists

of channel information, the action corresponds to resource allocation and positioning control, the reward reflects instantaneous coverage satisfaction, and the state transition is induced by channel dynamics and satellite motion. In this work, these components are designed as follows.

3.1.1 State

The state space presents the current circumstances or context in which an entity operates, capturing all relevant information that could impact the decision-making process. In this study, we define the state space based on the channel gain at time k as

$$s(k) \in \mathbb{S} = \{G_{LEO,h}(k), G_{LEO,u}(k), G_{LEO,g}(k), G_{h,u}(k), G_{h,g}(k), G_{u,g}(k)\}, \quad (11)$$

where $G_{LEO,h}(k)$, $G_{LEO,u}(k)$, $G_{LEO,g}(k)$, $G_{h,u}(k)$, $G_{h,g}(k)$, and $G_{u,g}(k)$ are the channel gains between the following entities: LEO satellite to HAPs, LEO satellite to UAVs, LEO satellite to GUs, HAPs to UAVs, HAPs to GUs, and UAVs to GUs, respectively. are the channel gains between the following entities: LEO satellite to HAPs, LEO satellite to UAVs, LEO satellite to GUs, HAPs to UAVs, HAPs to GUs, and UAVs to GUs, respectively. It is noted that the state is transformed into a flatten vector with $N = H(1+U) + G(1+H+U)$ elements. Here, the channels from the LEO satellite to HAPs ($G_{LEO,h}(k)$), UAVs ($G_{LEO,u}(k)$), and GUs ($G_{LEO,g}(k)$) comprise $(H+U+G)$ elements. The channels from HAPs to UAVs ($G_{h,u}(k)$) and GUs ($G_{h,g}(k)$) comprise $(HU+HG)$ elements. The channels from UAVs to GUs ($G_{u,g}(k)$) comprise (UG) elements. The system is modeled in discrete time slots, where the network state at each slot captures instantaneous channel conditions and network geometry, providing a slot-based MDP formulation for downlink coverage optimization.

3.1.2 Action

The action space represents the set of choices or decisions available to an agent in a specific state. In this study, the action space at time k is designed based on the controllable variables in problem (P1), which is given as

$$a(k) \in \mathbb{A} = \{L_h(k), L_u(k), L_{LEO,c}(k), L_{h,c}(k), L_{u,c}(k), b_{LEO}(k), b_h(k), b_u(k), p_{LEO}(k), p_h(k), p_u(k)\}. \quad (12)$$

The action is flattened to a vector with $M = 2((H+U) + (C+1)(1+H+U))$ elements. Here, the locations of HAPs ($L_h(k)$) and UAVs ($L_u(k)$) contribute $2(H+U)$ elements, representing position along the x- and y-axis. The locations of the beamforming centers for the LEO satellite

($L_{LEO,c}(k)$), HAPs ($L_{h,c}(k)$), and UAVs ($L_{u,c}(k)$) contribute $2C(1+H+U)$ elements, also representing changes on the x- and y-axis. The bandwidth allocation for the LEO satellite ($b_{LEO}(k)$), HAPs ($b_h(k)$), and UAVs ($b_u(k)$) contributes $(1+H+U)$ elements. The transmit power levels for the LEO satellite ($p_{LEO}(k)$), HAPs ($p_h(k)$), and UAVs ($p_u(k)$) contribute $(1+H+U)$ elements. This structure ensures that the action space comprehensively captures all relevant variables for decision-making.

3.1.3 Reward

The reward function in DRL acts as a feedback mechanism that indicates how successful or unsuccessful an action is within a given state. In this study, the reward function is designed based on the objective function of problem (P1) and is expressed as

$$r(k) = \begin{cases} \text{CON}(k), & \text{satisfied } C1-C4, \\ 0, & \text{otherwise.} \end{cases} \quad (13)$$

We assign 0 when constraints $C1 - C4$ are violated since infeasible actions cannot be executed in practice, this gate also avoids introducing an extra penalty scale that must be tuned relative to the coverage reward.

3.2 Actor and critic networks

The DAC algorithm consists of two deep neural network (DNN) components, known as the actor and critic networks [37], is adopted due to its ability to handle high-dimensional continuous action spaces and to provide stable learning through value-guided policy updates. These components in our work are described as follows.

3.2.1 Critic network

The critic network evaluates the selected action based on the current state using the state-action-value function (Q-function). This function estimates the expected future reward for taking a particular action $a(k)$ in a given state $s(k)$. The Q-function $Q(s(k), a(k); \mathbf{w}_c)$ in the critic network is designed as a DNN, where $\mathbf{w}_c$ represents the weights of the critic network. The weights of the critic network are updated using the temporal difference (TD) error, which serves as the loss function given as

$$L(\mathbf{w}_c) = \frac{1}{2} (Q^{\text{targ}} - Q(s(k), a(k); \mathbf{w}_c))^2. \quad (14)$$

Here Q^{targ} is the target value function, calculated as

$$Q^{\text{targ}} = r(k) + \gamma \max_{a \in \mathbb{A}} Q(s(k+1), a; \mathbf{w}_c), \quad (15)$$

where $r(k)$ is the reward received and $\gamma \in [0, 1]$ is the discount factor.

3.2.2 Actor network

The actor network is responsible for selecting actions $a(k)$ based on the current state $s(k)$ following the policy π . The policy $\pi(a(k)|s(k); \mathbf{w}_a)$ is designed using a DNN, where $\mathbf{w}_a$ is the weight of the actor network. The policy function is used to update the weight of actor network $\mathbf{w}_a$, which is given as

$$P(\mathbf{w}_a) = \mathbb{E}[\log \pi(a|s(k); \mathbf{w}_a) Q(s(k), a(k); \mathbf{w}_c)]. \quad (16)$$

This equation reflects the relationship between the actor and critic loss functions, which emphasizes the importance of updating the critic networks.

3.3 Temperature Function based Policy

The policy $\pi(a|s(k))$ determines the agent's action $a(k)$ based on the given states $s(k)$. This policy plays a crucial role in the agent's performance, learning efficiency, and adaptability to changes in the dynamic environment, particularly influencing training convergence. There are several policies, including the Epsilon-greedy policy [28], the SoftMax policy [16], and the Boltzmann exploration policy [3], among others. One commonly used policy is the ϵ -greedy policy, which balances exploration and exploitation by selecting a random action with probability ϵ and choosing the action that maximizes the expected reward with probability $1 - \epsilon$. The ϵ -greedy policy is expressed as

$$a(k) = \begin{cases} \text{random action} & \text{with prob. } \epsilon \\ \arg \max_{a \in \mathbb{A}} Q(s(k), a; \mathbf{w}_c) & \text{with prob. } 1 - \epsilon. \end{cases} \quad (17)$$

However, the ϵ -greedy policy can face challenges with convergence speed due to its reliance on uniformly random actions for exploration. To address this issue and reduce training time, an enhanced version called the decaying ϵ -greedy strategy has been developed. This approach aims to improve convergence by gradually decreasing the exploration probability over time, which in turn increases the probability of exploitation [19].

The decaying ϵ -greedy strategy can be expressed as

$$\epsilon(k+1) = \max\{\epsilon_{\min}, \epsilon(k)R_{\text{decay}}(k)\}, \quad (18)$$

where $\epsilon_{\min} \geq 0$ is the minimum probability for exploration, and $0 \leq R_{\text{decay}}(k) \leq 1$ is the decay function. While the decaying ϵ -greedy strategy accelerates convergence by gradually reducing the probability of exploration, it also increases the risk of the algorithm getting trapped in a local

optimum as the agent may no longer sufficiently explore alternative actions when ϵ becomes too low. To address this limitation and prevent repeated or suboptimal choices, we incorporate a temperature function in the proposed DAC algorithm, which is inspired by simulated annealing (SA) and generalized annealing principles [29, 30]. The purpose of this mechanism is to regulate exploration diversity in our high-dimensional continuous action space, where uniformly random exploration in conventional ϵ -greedy strategies can be inefficient and poorly scaled. This mechanism evaluates trial actions using the energy-based criterion and updates to preserve exploration diversity. From a theoretical perspective, annealing introduces a temperature schedule that encourages broader exploration at early stages and gradually shifts toward exploitation as the temperature decreases, which helps escape local optima in non-convex landscapes. This operation enables the agent to explore the solution space more broadly and escape local optima, especially during early training stages when the temperature is high. The expected impact is more stable convergence and improved performance compared to conventional method.

It should be noted that the ϵ -greedy policy serves as a baseline in the proposed DAC algorithm. However, the exploratory action selected with probability ϵ is not uniformly random as in conventional ϵ -greedy strategies, but is instead generated and accepted through a temperature-guided energy evaluation mechanism that explicitly accounts for exploration diversity and the structure of the action space.

Let $\tilde{a}(k)$ be the action determined by the temperature function. To determine $\tilde{a}(k)$, we first select a trial action $ta(k)$ randomly. The trial jump distance between the previous action ($a(k-1)$) and the current trial action ($\tilde{a}(k)$) is defined as $\Delta a(k) = \|\tilde{a}(k) - a(k-1)\|$ by using the distorted Cauchy-Lorentz visiting distribution [30], where $\|\cdot\|$ is the Euclidean norm. And we calculate the energy function $E(k)$, representing the difference in jump distances at time k and $k-1$, which is given by

$$E(k) = f_q(\Delta a(k)) - f_q(\Delta a(k-1)). \quad (19)$$

This mechanism allows the agent to favor action updates that preserve exploration diversity, thereby facilitating broader search of the solution space during early training stages. Here, the temperature function $f(\cdot)$ is defined based on the visiting parameter q as

$$f_q(\Delta a(k)) = \frac{(T_q(k))^{-\frac{d}{3-q}}}{[1 + (q-1) \frac{(\Delta a(k))^2}{2}]^{\frac{1}{q-1} + \frac{d-1}{2}} (T_q(k))^{\frac{d}{3-q}}}, \quad (20)$$

where $d = \dim(\mathbb{A})$ is the dimension of the action space, and $T_q(k) = T_q(0) \frac{2^{q-1}-1}{(1+k)^{q-1-1}}$ is the artificial temperature,

which decreases as time k increases. The visiting parameter q controls the exploration characteristics, where $q > 1$ generates heavy-tailed distributions for diverse exploration and q approaching 1 promotes exploitation.

The temperature function improves learning by evaluating energy difference $E(k)$ in (19), which measures exploration diversity changes. In our work, when $E(k) \geq 0$, $\tilde{a}(k)$ is selected as $\tilde{a}(k)$; otherwise, another trial action is randomly chosen until $E(k) \geq 0$ is satisfied. By doing this, the algorithm accepts actions that maintain exploration diversity, preventing premature convergence and providing more intelligent exploration than uniform random sampling in the standard exploration mechanism.

For exploitation, the next action is selected based on the maximum expected reward using the Q-value function with probability $1 - \epsilon$, which is given by $a(k) = \arg \max_{a \in \mathbb{A}} Q(s(k), a; \mathbf{w}_c)$.

To sum up, the proposed temperature function based policy update procedure, which differs from conventional ϵ -greedy exploration by generating exploratory actions through a temperature mechanism rather than uniform random sampling, is expressed as

$$a(k) = \begin{cases} \tilde{a}(k), & \text{with prob. } \epsilon, \\ \arg \max_{a \in \mathbb{A}} Q(s(k), a; \mathbf{w}_c), & \text{with prob. } 1 - \epsilon. \end{cases} \quad (21)$$

3.4 The operational procedure of the proposed DAC-temp algorithm

Algorithm 1 Pseudo code of the proposed DAC-Temp.

```

1: initialization:
2:   initial weight of actor network  $\mathbf{w}_a$ 
3:   initial weight of critic network  $\mathbf{w}_c$ 
4:   initial temperature function  $f_q$ 
5:   initial replay memory,  $\mathcal{D}$ 
6: while not converge:
7:   observe state  $s(k)$ 
8:   loop:
9:     random a trial action,  $\tilde{a}(k)$ 
10:    update temperature function,  $f_q(\Delta \tilde{a}(k))$ 
11:    calculate energy function,  $E(k)$ 
12:    if  $E(k) \geq 0$ :
13:      select action as  $a(k) = \tilde{a}(k)$ 
14:    break loop
15:  observe next state  $s(k+1)$ , and reward  $r(k)$ 
16:  store  $(s(k), a(k), r(k), s(k+1))$  in  $\mathcal{D}$ 
17:  if  $\mathcal{D}$  is full:
18:    compute target value
19:       $Q^{\text{targ}} = r(k) + \gamma \max_{a \in \mathbb{A}} Q(s(k+1), a; \mathbf{w}_c)$ 
20:    compute loss function
21:       $L(\mathbf{w}_c) = \frac{1}{2} (Q^{\text{targ}} - Q(s(k), a(k); \mathbf{w}_c))^2$ 
22:    compute policy function
23:       $P(\mathbf{w}_a) = \mathbb{E}[\log \pi(a|s(k); \mathbf{w}_a) Q(s(k), a(k); \mathbf{w}_c)]$ 
24:    update weight of actor and critic network
25:       $\mathbf{w}_c \leftarrow \mathbf{w}_c - \eta \nabla_{\mathbf{w}_c} L(\mathbf{w}_c)$ 
26:       $\mathbf{w}_a \leftarrow \mathbf{w}_a - \eta \nabla_{\mathbf{w}_a} J(\mathbf{w}_a)$ 

```

The procedure for the proposed DAC-Temp combined with the temperature function (DAC-Temp) is outlined as follows. Figure 3 illustrates the overall architecture of the proposed DAC-Temp framework, highlighting the temperature-guided action selection, experience storage, and network update process. First, an agent located on an LEO satellite is responsible for training the DAC-Temp model. The agent collects channel information from all communication links, including the channel gain from the LEO satellite to HAPs ($G_{LEO,h}(k)$), the LEO satellite to UAVs ($G_{LEO,u}(k)$), the LEO satellite to GUs ($G_{LEO,g}(k)$), HAPs to UAVs ($G_{h,u}(k)$), HAPs to GUs ($G_{h,g}(k)$), and UAVs to GUs ($G_{u,g}(k)$). This collected channel information serves as the input data, or state ($s(k)$), for the DAC-Temp model. After that, the agent interacts with the environment by taking an action using the temperature function based policy in (21). With the current state ($s(k)$) and the selected action, the agent observes the next state ($s(k+1)$) and the reward ($r(k)$), and stores them in the replay memory ($\mathcal{D}$). When the replay memory ($\mathcal{D}$) is full, the agent trains the DAC-Temp model following the training process of the DAC algorithm by computing the target value (Q^{targ}), the policy function ($P(\mathbf{w}_a)$), and the loss function ($L(\mathbf{w}_c)$). Finally, the weights of the actor and critic networks ($\mathbf{w}_a$ and $\mathbf{w}_c$) are updated using the stochastic gradient descent (SGD) algorithm with the learning rate η , until convergence. Algorithm 1 shows the details of the pseudo code for the proposed DAC-Temp algorithm.

3.5 Comparative algorithms for SAGIN coverage optimization

To benchmark the effectiveness of our proposed DAC-Temp algorithm, we compare it with five alternative algorithms: DAC, SAC, DDPG, GA, and GS. These methods are adapted to solve the same joint optimization problem. DAC is included as the closest structural baseline to the proposed method, sharing the same actor-critic architecture but without the temperature-guided exploration mechanism, which provides the direct impact of the temperature function to be isolated. SAC and DDPG are selected as representative actor-critic DRL methods with different policy learning and exploration strategies, serving to verify that the proposed DAC-Temp outperforms alternative learning-based control approaches. In contrast, GA and GS represent classical population-based and gradient-based optimization techniques that are widely used for non-convex problems, providing non-learning baselines for assessing performance gains and scalability.

3.5.1 Deep actor-critic

This method follows the same actor-critic architecture as our proposed DAC-Temp algorithm. The structural design of

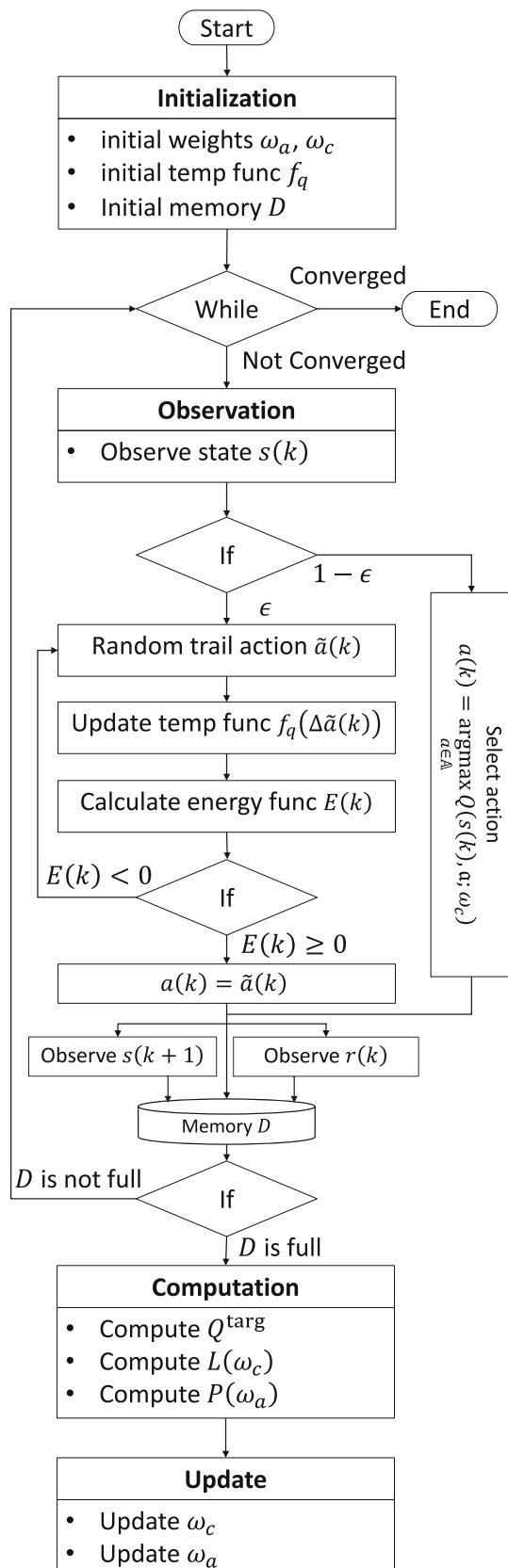


Fig. 3 Flowchart of the proposed DAC-Temp algorithm

DAC includes two core components: an actor network and a critic network. The actor network maps the observed environment state to action, which determines the positions of aerial nodes, beamforming directions, bandwidth allocation, and transmit power. The critic network evaluates these actions by estimating the value function, which guides the learning of the actor. Unlike DAC-Temp, which uses a temperature-based mechanism for adaptive exploration, this baseline DAC implementation relies on a conventional ϵ -greedy strategy. The ϵ value decays over time to balance exploration and exploitation during training. The state and action spaces and reward formulation remain consistent with those defined in Section 3. DAC updates its policy through advantage estimation derived from the critic's feedback, enabling the agent to iteratively refine its decisions by interacting with the environment. This architecture allows DAC to jointly optimize multiple system parameters while learning directly from experience.

3.5.2 Soft actor-critic

SAC is a DRL algorithm that uses a stochastic policy to explore the action space. Unlike DAC, which outputs deterministic actions, SAC models its policy as a probability distribution over actions [8]. At each decision step, actions are sampled from this distribution rather than directly selected, allowing the agent to maintain exploration throughout training. In this work, the SAC architecture consists of an actor network, two Q-value networks (critics), and a separate value network. The actor network learns a stochastic policy by outputting the parameters of a probability distribution. The two Q-value networks help mitigate overestimation bias by selecting the minimum predicted value during updates. The value network estimates the expected return of the current state under the policy. All networks are updated using experience replay and soft target updates. SAC optimizes an entropy-augmented objective function that encourages the policy to maximize both reward and entropy. This entropy term helps exploration by allowing more diverse action selection. The key structural difference from DAC lies in SAC's use of this maximum entropy framework, which simultaneously maximizes expected return and randomness in action selection. While this strategy improves performance in complex environments with many optimal policies, it may lead to slower convergence in simpler or low-variance tasks due to sustained exploration [22].

3.5.3 Deep deterministic policy gradient

DDPG is an actor-critic algorithm developed for continuous control tasks. In this work, the structural design of DDPG consists of four neural networks: an actor network that deterministically maps states to actions, a critic network

that estimates Q-values, and their respective target networks that are used to stabilize training. These target networks are updated slowly to reduce the variance in value estimates and prevent instability [35]. Unlike DAC, which uses an ϵ -greedy strategy for exploration, DDPG introduces temporally correlated noise to the action output during training using the Ornstein-Uhlenbeck process. This allows smoother exploration trajectories. The actor network is trained to maximize the expected Q-value predicted by the critic, and the critic is updated by minimizing the temporal difference (TD) error using experience replay. While DDPG can achieve strong performance in high-dimensional continuous spaces, it is often more sensitive to hyperparameters and exploration quality, which can limit its robustness in complex or noisy environments [2].

3.5.4 Genetic algorithm

GA is a population-based metaheuristic inspired by the process of natural selection [15]. We aim to optimize the coverage of SAGIN by jointly controlling the location of aerial devices, transmit power, beamforming direction, and bandwidth allocation. To address this problem, we implement GA, which provides an evolutionary approach to finding near-optimal solutions. The GA implementation for our SAGIN coverage optimization is detailed as follows [1].

Step-1) Initialization: A population of size Z is randomly initialized. Each individual solution $z \in Z$ encodes a candidate action a_z .

Step-2) Fitness Evaluation: Each action is evaluated based on the reward $r(t)$, which reflects the percentage of connected GUs satisfying the data rate threshold.

Step-3) Selection: A selection mechanism (tournament selection) is used to probabilistically favor high-reward actions for reproduction.

Step-4) Crossover: Selected actions are combined to form new offspring solutions.

Step-5) Mutation: To maintain diversity and avoid local optima, each gene in the offspring action a_z is perturbed with a small mutation probability μ . This process repeats over multiple generations until convergence. GA is robust and requires no gradient information, making it well-suited for high dimensional black-box optimization. However, it may require a large number of evaluations to find optimal or near-optimal solutions.

3.5.5 Gradient search

GS is a local optimization method that relies on computing the gradient of the reward function with respect to the controlled parameters. In our case, it works as follows.

Step-1) Initialization: The process starts with a randomly initialized vector $a_{z,d} \in a_z$ where d is the element of action.

Step-2) Gradient Computation: At iteration w , the gradient of the reward function $r^t(a_z)$ with respect to each parameter is approximated using finite differences

$$\frac{\partial r(t)}{\partial a_{z,d}} \approx \frac{r^t(a_z + \kappa \mathbf{e}_d) - r^t(a_z)}{\kappa}$$

where κ is a small constant and $\mathbf{e}_d$ is the unit vector in the d -th direction.

Step-3) Update Step: The action vector is updated via

$$a_z^{(w+1)} = a_z^{(w)} + \eta \nabla r^t(a_z^{(k)}) + \eta \left(\nabla r^t(a_z^{(w)}) - \lambda \nabla \Phi(a_z^{(w)}) \right)$$

where η is the learning rate, λ is the penalty imposed by the log-barrier and $\Phi(a_z^{(w)})$ is the log-barrier penalty function to maintain feasibility [25].

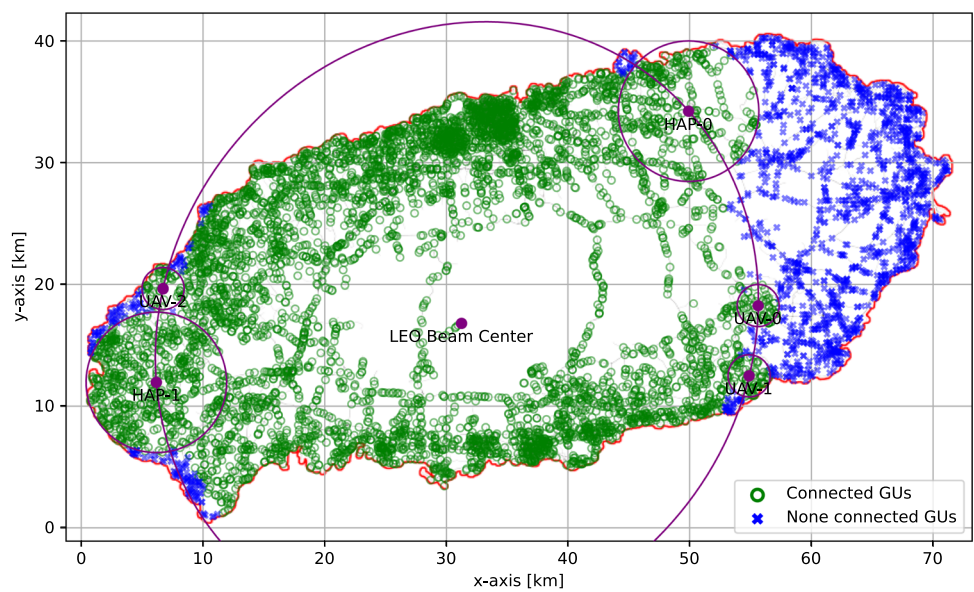
Step-4) Convergence Check: Stop if the improvement is below a predefined threshold $|r^t(a_z^{(w+1)}) - r^t(a_z^{(w)})| < \delta$ or a maximum number of iterations W is reached.

3.6 Time complexity analysis

The proposed DAC-Temp model consists of two networks: the actor network and the critic network. In the deployment process, the time complexity is determined by the actor network, as the critic network is no longer used to evaluate actions. The actor network is designed with N neurons in the input layer, corresponding to the number of elements in the state, N neurons in each of the L hidden layers, and M neurons in the output layer representing the number of elements in the action. All layers are fully connected. According to matrix computations [24], the feed-forward calculations for the actor network require $N_{\text{in}} \times N_{\text{hid},1} \times (L-1) \times N_{\text{hid},L} \times N_{\text{hid},L} \times M_{\text{out}}$ computation steps for the input (in.), hidden (hid.), and output (out.) layers, respectively. Therefore, the time complexity for deploying the DAC-Temp model is $O(N^2 + (L-1)N^2 + NM)$. Since $N \gg M$, we can simplify the time complexity of the proposed DAC-Temp as $O(LN^2)$. The proposed DAC-Temp follows an offline training and online deployment paradigm. Training is performed offline, while online deployment uses only the trained actor network to generate actions through a single feed-forward inference, which has low latency and support real-time control.

DDPG and SAC are actor-critic algorithms with a structure similar to DAC-Temp. Both use an actor network and one or more critic networks. During deployment, only the actor is used. The main difference lies in the number of critic networks and the update rules during training. Since deployment only relies on the actor network, the inference complexity of DDPG and SAC is also $O(LN^2)$, assuming a similar architecture size.

Fig. 4 Optimized coverage of GUs in Jeju Island under DAC-Temp. Green circles denote connected GUs (meeting the 1 Mbps threshold), blue crosses denote disconnected GUs, and the LEO beam center, HAPs, and UAVs are indicated. DAC-Temp places HAPs/UAVs near the boundary of LEO coverage to reinforce dense, weak-link regions. Some edge users remain disconnected due to limited bandwidth



GS is a deterministic optimization method that iteratively adjusts control parameters to maximize the objective. The computational complexity of GS depends on the number of parameters being updated and the number of iterations required for convergence. Let D denote the total number of parameters and W be the number of iterations required to reach a given convergence threshold δ . In each iteration, computing the gradient and updating parameters incurs a cost of $O(D)$. Hence, the total computational complexity of GS can be represented as $O(T \times D)$.

GA is a population-based metaheuristic inspired by biological evolution. It explores the search space by applying selection, crossover, and mutation operators to a population of candidate solutions. Let Z denote the population size, W the number of generations, and D the number of control parameters (decision variables). Evaluating the fitness function for each candidate requires $O(D)$ leads to a per-generation complexity of $O(Z \times D)$. Therefore, the overall complexity of GA across all generations is $O(Z \times W \times D)$.

4 Performance evaluations

In this section, we evaluate the performance of the proposed DAC-Temp algorithm against several baseline methods for SAGIN coverage optimization. These include three DRL algorithms such as DAC, DDPG, and SAC, as well as two traditional optimization methods including GA and GS. All methods are tested under the same simulation setup based on a realistic urban-scale environment over Jeju Island, South Korea. This comprehensive comparison allows us to assess the effectiveness and robustness of DAC-Temp in maximizing user coverage while satisfying system constraints. In this

simulation, Jeju Island is represented as a polygon Ω , which is approximated using publicly available satellite and street imagery from Google Maps. The boundary of the island is manually outlined to construct the simulation region. GUs are synthetically generated along the extracted road network using a Poisson point process, with up to 8000 GUs considered to represent dense deployment scenarios. This approach preserves key geographic features and realistic GU clustering constrained by actual street layouts, rather than assuming uniform spatial distributions. Figure 4 illustrates the constructed polygonal region of Jeju Island and the density heatmap of GUs distributed along the road network. The simulations are performed on a computer equipped with an AMD Ryzen 5 3600 CPU and an NVIDIA GeForce RTX 2080 Ti GPU.

Table 2 Simulation environment setup

Parameters	Value
Altitude of LEO satellite, Z_{LEO}	340 Km
Altitude of HAPs, Z_{HAP}	10 Km
Altitude of UAVs, Z_{UAV}	1 Km
Number of HAPs, H	2 devices
Number of UAVs, U	3 devices
Polygon Region, Ω	Jeju Island
Maximum antennas gain of transmitters, $G_{\max}$	30 dB
Maximum power of transmitters, $(p_{\max}^{LEO}, p_{\max}^{HAP}, p_{\max}^{UAV})$	32 dBm
3dB angle of LEO satellite, $\theta_{LEO, 3dB}$	1 Degree
3dB angle of HAPs, $\theta_{h, 3dB}$	10 Degree
3dB angle of UAV, $\theta_{u, 3dB}$	30 Degree
Noise power AWGN, σ_A^2	-100 dBm
Minimum data rate threshold, $R_{\min}$	1 Mbps

Table 2 provides a summary of the simulation environment and configuration.

The proposed DAC-Temp model employs the Rectified Linear Unit (ReLU) activation function for all neurons in the hidden layers of both the actor and critic networks. ReLU is widely used in neural networks due to its simplicity and effectiveness in handling the vanishing gradient problem. To update the weights of all neurons, we use stochastic gradient descent (SGD) with backpropagation, which adjusts network weights incrementally based on the gradient of the loss function. To ensure fair comparison, all DRL baselines including DAC, SAC, and DDPG use identical hyperparameters including learning rate, discount factor, network architecture, and initialization settings as shown in Table 3, allowing the effect of the temperature-based exploration mechanism to be isolated. The same random seeds are used across all DRL methods to eliminate initialization bias, all methods adopt the same decaying ϵ -greedy exploration strategy, and each method is evaluated over multiple independent runs to assess statistical robustness.

Figure 4 illustrates the optimized beam coverage achieved by the DAC-Temp algorithm when serving 8000 GUs across Jeju Island. In the figure, green circles represent GUs that are successfully connected, each receiving at least the minimum data rate of 1 Mbps, while blue crosses indicate GUs that could not be served due to limited resources. The HAPs and UAVs are distributed toward the outer regions of the LEO satellite beam coverage. This placement strategy demonstrates how the DAC-Temp algorithm dynamically adapts to GU density by placing transceivers where they are most needed typically near the boundaries of LEO's direct coverage, where GU density is high but link quality is weaker. By optimizing both node placement and resource allocation, the

Table 3 DRL algorithms setup

Parameters	Value
Learning rate, η	0.0001
Discount factor, γ	0.99
Replay memory size	10000
Batch size	1024
Number of hidden layer for actor network	10 layers
Number of hidden layer for critic network	10 layers
Activation function	ReLU
Optimizer	SGD
Soft update factor, τ	0.005
Initial exploration, ϵ_0	0.20
Minimum exploration, $\epsilon_{\min}$	0.05
Exploration decay rate, R_{decay}	0.995
Training episodes	1000
Device	GPU

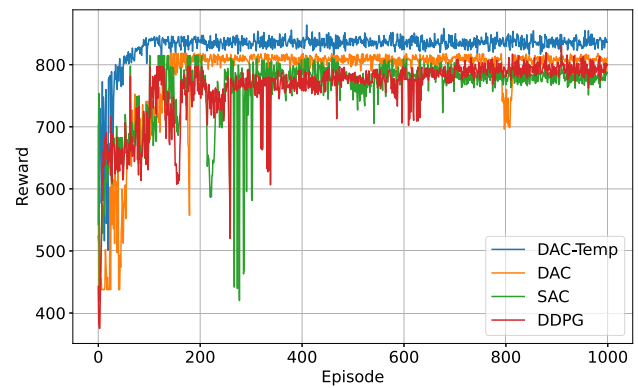


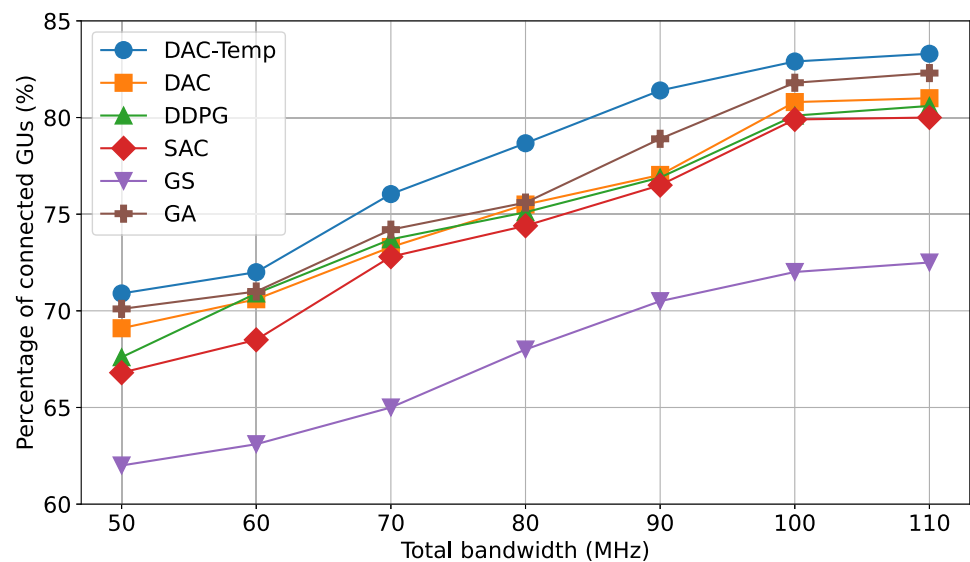
Fig. 5 Convergence performance of DAC-Temp compared to DAC, SAC, and DDPG: episodic reward versus training episodes. DAC-Temp achieves higher reward and stabilizes within 160 episodes, indicating faster and more stable learning

algorithm significantly improves overall connectivity. However, due to the large number of GUs and limited bandwidth, some GUs located at the distant edges remain disconnected. This outcome highlights the practical limitations of the system under constrained capacity, while also showcasing the effectiveness of the proposed algorithm in maximizing coverage under high-demand conditions.

Figure 5 presents the convergence behavior of the proposed DAC-Temp algorithm compared to three baseline DRL methods: DAC, SAC, and DDPG. The y-axis represents the episodic reward obtained during training, while the x-axis shows the number of training episodes. As observed, DAC-Temp consistently achieves higher rewards across episodes and converges faster than the other methods. The reward curve of DAC-Temp stabilizes within the first 160 episodes, which indicates improved learning efficiency and reduced variance during exploration. In contrast, DAC, SAC, and DDPG show slower convergence and greater fluctuations in reward, particularly in early training stages. This highlights the benefit of incorporating temperature-based exploration, which helps DAC-Temp to balance exploration and exploitation more effectively during learning.

Figure 6 shows the performance of various optimization algorithms in connecting 1000 GUs, measured by the percentage of GUs successfully connected across a bandwidth range of 50 MHz to 110 MHz. Here the number of generated GUs is fixed in this experiment, and only the total bandwidth is varied. As the total bandwidth increases, all algorithms show improved GUs connectivity, highlighting the importance of bandwidth as an important resource in coverage optimization. DAC-Temp consistently outperforms the other algorithms, achieving the highest percentage of connected users at each bandwidth level. This suggests that its temperature-based exploration mechanism enables better policy learning for optimizing user coverage. Other

Fig. 6 The performance of various optimization algorithms in connecting 1000 GUs, measured by the percentage of GUs successfully connected across a bandwidth range of 50 MHz to 110 MHz



DRL-based methods perform similarly and exhibit steady improvements with increasing bandwidth, although their gains are slightly lower than DAC-Temp. GA shows competitive performance, especially at higher bandwidths, where it closely matches the DAC-Temp. In contrast, GS performs significantly worse across all bandwidth levels. The reason lies in its deterministic, gradient-based nature, which makes it prone to getting trapped in local minima in high-dimensional spaces like our problem. Furthermore, GS lacks the exploratory behavior necessary to escape suboptimal solutions.

Figure 7 shows how the total data rate changes as the number of generated GUs increases, with the bandwidth set to 100 MHz, the total bandwidth is fixed at 100 MHz

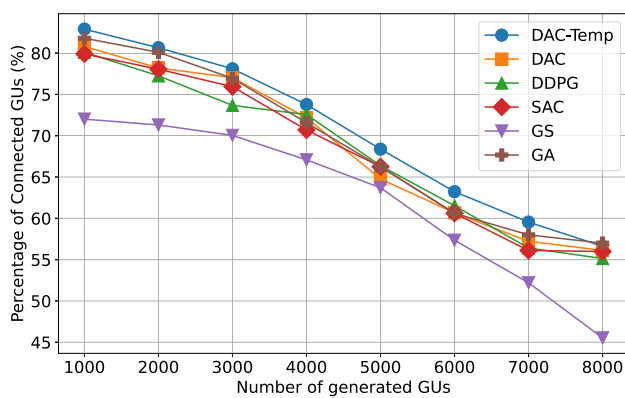


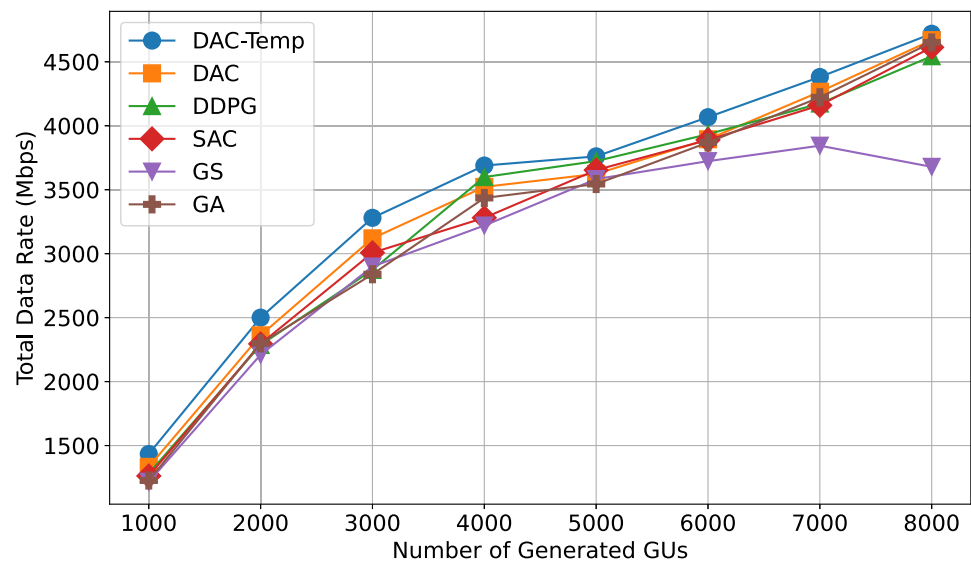
Fig. 7 Percentage of connected GUs versus the number of generated GUs (1000-8000) at a fixed total bandwidth of 100 MHz. DAC-Temp maintains the highest connectivity as user density increases. GA plateaus beyond around 6000 GUs. GS degrades rapidly and DRL-based baselines degrade less but remain below DAC-Temp

in this experiment, and only the number of generated GUs is varied. As user density increases, the network becomes more congested, which makes it harder to maintain consistent connectivity across all GUs. Among all the methods, DAC-Temp consistently achieves the highest connectivity, as its temperature-guided exploration enables adaptive refinement of control actions under increasing GU density. GA maintains competitive performance up to 6000 GUs but eventually stops improving due to its static population-based search, while GS degrades rapidly as GU density increases, reflecting its limitations in adaptability and scalability. In contrast, DRL-based methods handle growing complexity by continuously adjusting to dynamic environments. Overall, the figure shows that while all algorithms degrade under heavier loads, DAC-Temp maintains the highest level of connectivity, which makes it more suitable for dense and dynamic SAGIN environments.

Figure 8 illustrates how the total data rate changes as the number of generated GUs increases, with the bandwidth set to 100 MHz. In this experiment, the total bandwidth is fixed to 100 MHz, and the number of GUs is varied to evaluate throughput scaling under denser GU deployments. The proposed method, particularly DAC-Temp, consistently achieves higher data rates across all user scales due to its adaptive learning capabilities. Traditional methods like GA perform well but begin to lag slightly as the number of users grows. In contrast, GS shows the lowest performance at higher user densities due to its local search behavior and limited adaptability. Overall, DRL methods demonstrate better scalability and throughput in dense SAGIN scenarios.

Table 4 reports the mean performance, variance, and 95% confidence intervals (CI) over 100 independent runs. Heuristic methods such as GA and GS exhibit lower variance due

Fig. 8 Total data rate versus the number of generated GUs at a fixed total bandwidth of 100 MHz. DAC-Temp sustains the highest throughput across scales. GA performs well but lags as the number of GUs grows. GS performs worst at high densities. DRL methods show better scalability overall



to their deterministic or population-based search processes. However, GS converges to suboptimal solutions, while GA, although competitive in terms of performance quality, suffers from limited scalability and long execution time in dynamic, high-dimensional SAGIN environments. In contrast, DRL-based methods inherently exhibit higher variance due to stochastic exploration. The proposed DAC-Temp achieves a favorable balance by reducing variance relative to standard DRL baselines, demonstrating improved stability and consistent performance which benefits from the temperature-based exploration mechanism.

5 Conclusion

In this paper, we addressed the challenge of optimizing coverage in SAGIN by jointly adjusting the placement of HAPs and UAVs, transmit power, bandwidth allocation, and beam-forming direction. To solve the problem, we proposed a DAC algorithm combined with a temperature-based exploration mechanism (DAC-Temp) to improve exploration and reduce training time. To validate our approach, we conducted simulations and compared DAC-Temp against several baseline

methods, including two traditional optimization techniques such as GA and GS as well as several DRL algorithms, namely standard DAC, SAC, and DDPG. The experimental results conducted over Jeju Island, South Korea, showed that DAC-Temp consistently outperformed all baseline methods in terms of coverage, convergence speed, and bandwidth efficiency due to its adaptive temperature-based exploration mechanism that effectively handles increasing GU density and dynamic network conditions. Overall, this study presents a scalable and efficient solution for improving global communication in SAGIN networks. By leveraging intelligent control of aerial nodes and advanced reinforcement learning techniques, our work contributes to the development of adaptive, high-performance communication infrastructure.

For future work, we plan to explore more dynamic network environments by incorporating GU and aerial platform mobility, as well as mobility-induced handover mechanisms under varying traffic demands. In addition, we plan to extend the framework to support multi-agent collaboration between aerial platforms, which enables decentralized learning and decision-making. Integrating energy constraints, fairness metrics, and real-time traffic prioritization is also promising directions to further enhance the practical applicability of the proposed approach in next-generation SAGIN.

Table 4 Statistical performance comparison over 100 independent runs

Method	Mean (%)	Variance	95% CI
DAC-Temp	83.0	1.64	[82.75, 83.25]
DAC	80.9	2.19	[80.61, 81.19]
DDPG	80.1	2.36	[79.80, 80.40]
SAC	79.8	2.55	[79.47, 80.13]
GA	81.8	0.86	[81.65, 81.95]
GS	72.0	0.75	[71.87, 72.13]

Author Contributions Credit Author Statement Vitou That: Investigation, Software, Validation, Formal analysis, Writing — review & editing, Methodology, Visualization, Investigation. Sengly Muy: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing — original draft, Investigation, Validation. Jung-Ryun Lee: Conceptualization, Methodology, Validation, Funding acquisition, Project administration, Supervision.

Funding This work was supported in part by the Ministry of Science and ICT (MSIT), Korea through Information Technology Research Center (ITRC) Support Program under Grant IITP-RS-2025-00436248 Supervised by the Institute for Information & Communications Technology

Planning & Evaluation(IITP), and in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MEST) under Grant RS-2023-00251105.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

References

- Bouazid SE, Seresstou Y, Raoof K et al (2020) Moonga: multi-objective optimization of wireless network approach based on genetic algorithm. *IEEE Access* 8:105793–105814
- Cao Z, Zhou P, Li R, et al (2020) Multiagent deep reinforcement learning for joint multichannel access and task offloading of mobile-edge computing in industry 4.0. *IEEE Internet Things J* 7(7):6201–6213
- Cesa-Bianchi N, Gentile C, Lugosi G, et al (2017) Boltzmann exploration done right. *Adv Neural Inf Process Syst* 30
- Cui H, Zhang J, Geng Y et al (2022) Space-air-ground integrated network (sagin) for 6g: Requirements, architecture and challenges. *China Commun* 19(2):90–108
- Dai C, Zheng G, Chen Q (2018) Satellite constellation design with multi-objective genetic algorithm for regional terrestrial satellite network. *China Commun* 15(8):1–10
- Fan K, Feng B, Zhang X et al (2022) Network selection based on evolutionary game and deep reinforcement learning in space-air-ground integrated network. *IEEE Trans Netw Sci Eng* 9(3):1802–1812
- Gao Q, Jia M, Guo Q, et al (2023) Jointly optimized beam-forming and power allocation for full-duplex cell-free noma in space-ground integrated networks. *IEEE Trans Commun*
- Huang C, Chen G, Xiao P, et al (2024) Joint offloading and resource allocation for hybrid cloud and edge computing in sagins: A decision assisted hybrid action space deep reinforcement learning approach. *IEEE J Sel Areas Commun*
- Kodheli O, Lagunas E, Maturro N et al (2020) Satellite communications in the new space era: A survey and future challenges. *IEEE Commun Surv Tutor* 23(1):70–109
- Li Y, Hu X, Zhuang Y et al (2019) Deep reinforcement learning (drl): Another perspective for unsupervised wireless localization. *IEEE Internet Things J* 7(7):6279–6287
- Luong NC, Hoang DT, Gong S et al (2019) Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Commun Surv Tutor* 21(4):3133–3174
- Lv X, Wu S, Li A et al (2022) A weighted graph-based handover strategy for aeronautical traffic in leo satcom networks. *IEEE Netw Lett* 4(3):132–136
- Maral G, Bousquet M, Sun Z (2020) Satellite communications systems: systems, techniques and technology. John Wiley & Sons
- McDowell JC (2020) The low earth orbit satellite population and impacts of the spacex starlink constellation. *Astrophys J Lett* 892(2):L36
- Mehboob U, Qadir J, Ali S et al (2016) Genetic algorithms in wireless networking: techniques, applications, and issues. *Soft Comput* 20:2467–2501
- Mei J, Xiao C, Szepesvari C, et al (2020) On the global convergence rates of softmax policy gradient methods. In: International conference on machine learning, PMLR, pp 6820–6829
- Mekikis PV, Antonopoulos A, Kartsakli E, et al (2016) Connectivity analysis in wireless-powered sensor networks with battery-less devices. In: 2016 IEEE global communications conference (GLOBECOM), IEEE, pp 1–6
- Mollahasani S, Erol-Kantarci M, Hirab M et al (2021) Actor-critic learning based qos-aware scheduler for reconfigurable wireless networks. *IEEE Trans Netw Sci Eng* 9(1):45–54
- Morales M (2020) Grokking deep reinforcement learning. Manning Publications
- Ouyang Q, Ye N, An J (2025) On the vulnerability of mega-constellation networks under geographical failure. *IEEE Trans Netw*
- Pan H, Liu Y, Sun G, et al (2025) Cooperative uav-mounted riss-assisted energy-efficient communications. *IEEE Trans Mobile Comput*
- Qin Y, Yang Y, Tang F, et al (2024) Differentiated federated reinforcement learning based traffic offloading on space-air-ground integrated networks. *IEEE Trans Mobile Comput*
- Qu L, Xu G, Zeng Z et al (2022) Uav-assisted rf/fso relay system for space-air-ground integrated network: A performance analysis. *IEEE Trans Wireless Commun* 21(8):6211–6225
- Sedgewick R, Wayne K (2016) Computer science: An interdisciplinary approach. Addison-Wesley Professional
- Sun W, Yuan YX (2006) Optimization theory and methods: nonlinear programming, vol 1. Springer Science & Business Media
- Tang F, Zhou Y, Kato N (2020) Deep reinforcement learning for dynamic uplink/downlink resource allocation in high mobility 5g hetnet. *IEEE J Sel Areas Commun* 38(12):2773–2782
- Tang F, Hofner H, Kato N et al (2021) A deep reinforcement learning-based dynamic traffic offloading in space-air-ground integrated networks (sagin). *IEEE J Sel Areas Commun* 40(1):276–289
- Tokic M, Palm G (2011) Value-difference based exploration: adaptive control between epsilon-greedy and softmax. In: Annual conference on artificial intelligence, Springer, pp 335–346
- Van Laarhoven PJ, Aarts EH, van Laarhoven PJ et al (1987) Simulated annealing. Springer
- Xiang Y, Gubian S, Suomela B et al (2013) Generalized simulated annealing for global optimization: the gensa package. *R J* 5(1):13
- Xu C, Xiang L, An J et al (2022) Otfis-aided ris-assisted sagin systems outperform their ofdm counterparts in doubly selective high-doppler scenarios. *IEEE Internet Things J* 10(1):682–703
- Xu H, Han S, Li X et al (2023) Anomaly traffic detection based on communication-efficient federated learning in space-air-ground integration network. *IEEE Trans Wireless Commun* 22(12):9346–9360

33. Yang H, Xie X (2019) An actor-critic deep reinforcement learning approach for transmission scheduling in cognitive internet of things systems. *IEEE Syst J* 14(1):51–60
34. Zhang P, Su Y, Wang J et al (2022) Reinforcement learning assisted bandwidth aware virtual network resource allocation. *IEEE Trans Netw Serv Manage* 19(4):4111–4123
35. Zhang S, Liu A, Han C et al (2023) Multiagent reinforcement learning-based orbital edge offloading in sagin supporting internet of remote things. *IEEE Internet Things J* 10(23):20472–20483
36. Zhang Z, Zhang J, Zhang Y et al (2023) Deep reinforcement learning based dynamic beam selection in dual-band communication systems. *IEEE Trans Wireless Commun* 23(4):2591–2606
37. Zhong C, Lu Z, Gursay MC et al (2019) A deep actor-critic reinforcement learning framework for dynamic multichannel access. *IEEE Trans Cognit Commun Netw* 5(4):1125–1139

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.