SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2026, Vol. 69, Iss. 9, 190304:1–190304:22
https://doi.org/10.1007/s11432-025-5063-7

Special Topic: User-Centric Intelligent Networking

LLM-enhanced-EA-driven automated optimization framework for joint indoor deployment of multiple BSs and RISs

Sichong LIAO[1], Jinbo HOU[1,2], Yi ZHAO[3], Ziming LIU[1], Haonan HU[1*], Zeyang LI[4], Kezhi WANG[2] & Jie ZHANG[5,6*]

[1]*School of Electronic and Electrical Engineering, The University of Sheffield, Sheffield S10 2TN, UK*
[2]*Department of Computer Science, Brunel University of London, Middlesex UB8 3PH, UK*
[3]*Department of Information Technology, Uppsala University, Uppsala 75105, Sweden*
[4]*School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[5]*School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm 11428, Sweden*
[6]*R&D Department, Ranplan Wireless Network Design Ltd., Cambridge CB23 3UY, UK*

Received 10 October 2025/Revised 15 March 2026/Accepted 29 July 2026/Published online 19 August 2026

Abstract The joint deployment of base stations (BSs) and reconfigurable intelligent surfaces (RISs) is considered a promising but underexplored solution to simultaneously improve indoor coverage performance and reduce network costs. Classical evolutionary algorithms (EAs) can solve this bi-objective non-convex mixed integer programming problem. However, EAs often suffer from efficiency issues due to convergence on local optima, and require redesigning the mechanism to concentrate on the fruitful areas of the search space, especially when the space is large. In this work, we use the large language model (LLM)-enhanced non-dominated sorting genetic algorithm II (NSGA-II) to directly solve the bi-objective optimization problem, assisted by an LLM to enhance its concentrated exploration capability by extending the exploration in language space. To minimize manual engineering, we designed an automated LLM-based module that generates tailored optimization prompts from templates, user descriptions, and reference materials. Simulations show that LLM-NSGA-II, guided by these automated prompts, matches the performance of manual engineering. When applied to joint BS and RIS deployment, LEAO achieves a cost efficiency of $55.7\% \pm 7.9\%$ across five independent runs, exceeding the strongest baseline by 11.2 percentage points and indicating a more favorable balance between deployment cost and indoor coverage.

Keywords optimization, evolutionary algorithm (EA), NSGA-II, intelligent operator, large language model (LLM), prompt engineering, base station (BS), reconfigurable intelligent surface (RIS)

Citation Liao S C, Hou J B, Zhao Y, et al. LLM-enhanced-EA-driven automated optimization framework for joint indoor deployment of multiple BSs and RISs. Sci China Inf Sci, 2026, 69(9): 190304, https://doi.org/10.1007/s11432-025-5063-7

1 Introduction

1.1 Background and motivation

The radio access network (RAN) constitutes approximately 65%–70% of the total network cost in mobile networks such as 5G[1)]. State-of-the-art RAN planning and optimization software tools can achieve cost reductions of up to 30% while meeting stringent performance requirements. As reported in studies[2)] [1], data traffic is expected to increase substantially by 2030, and more than 80% of data traffic occurs indoors. Therefore, efficient indoor RAN planning and optimization are crucial to reduce both capital expenditures (CAPEX) and operational expenditures (OPEX) while ensuring the network coverage requirement to realize capacity enhancement. Recently, the joint deployment of base stations (BSs) and reconfigurable intelligent surfaces (RISs) has been considered as a cost-effective solution to improve the indoor coverage performance. An RIS is generally composed of a large number of passive elements, each being able to reflect the incident signal, creating additional line-of-sight (LOS) paths. Therefore, the indoor coverage performance is enhanced with lower network costs [2].

* Corresponding author (email: huhn@cqupt.edu.cn, Jie.Zhang@ranplanwireless.com)

1) O-RAN Alliance. O-RAN: towards an open and smart RAN. 2018. Accessed: 04-Oct-2025.
2) Huawei. Five trends to small cell 2020. 2016. Accessed: 04-Oct-2025.

There have already been some studies investigating how to optimize the joint deployment of BSs and RISs. In [3], the authors optimized the placement of an RIS using a trained neural network to maximize the average fractional increase in coverage as the single objective. In [4], a deep reinforcement learning (DRL) algorithm is given to find optimized energy efficiency as a single objective by designing a single RIS position. However, these studies only considered joint deployment with a single RIS. Based on these studies, the authors in [5] jointly optimized multiple RIS locations using a partial enumeration method to minimize the total cost of RIS deployment subject to constraints on the signal-to-noise ratio (SNR) performance. In [6], the authors tried both a continuous relaxation with block coordinate ascent (BCA) and a model-free reinforcement learning method to minimize the number of deployed RISs while maximizing the SNR evaluated by Jain's fairness index. Based on gradient descent, the authors in [7] optimized the locations of RISs to maximize a coverage performance function as a single objective, reflecting the anisotropic coverage performance for the joint deployment. These studies focus on optimizing their formulated objective function as a single objective and on the distributed deployment of multiple RISs with a fixed RIS count, thus not considering the optimization for the number of RISs. The research in [8] optimized the locations of RISs using a genetic algorithm (GA) to minimize access point transmit energy subject to constraints on the received SNR requirement. In particular, it obtained the optimal number of RISs by enumeration. In [9], the authors also obtained the optimal number of RISs by enumeration while optimizing the locations of RISs subject to a minimum average achievable rate threshold via a differential evolution algorithm. Nevertheless, to reduce the total cost of any given joint deployment of BSs and RISs while satisfying coverage performance requirements, the number of BSs and RISs should be simultaneously optimized to ensure a convincing cost-effective deployment. Although the work in [10, 11] obtained the optimal number configurations for both BSs and RISs, both the partial enumeration and the full brute-force method for searching the number settings of BSs and RISs are the key obstacles to efficient joint deployment.

However, despite the extensive literature discussed above, the optimal joint deployment of multiple BSs and RISs necessitates further investigation for the following three reasons.

• **Complex mixed-integer search space:** Efficiently optimizing the joint selection of (i) BS/RIS locations and (ii) the number of deployed units remains an open problem. This leads to a complex non-convex mixed-integer programming (MIP) problem [5, 9], where discrete placement variables are coupled with continuous performance metrics. The complex search space possesses sparse deployment solutions with cost-effective performance, making partial enumeration, greedy methods, or brute-force search computationally prohibitive.

• **Cost-performance trade-off:** The optimal solutions for joint BS/RIS deployment are still insufficiently explored. The design must simultaneously (i) minimize total network cost and (ii) maximize coverage or capacity, resulting in a bi-objective combinatorial optimization problem. Most existing studies [3–7] assume a fixed number of RISs, excluding cost from optimization. Others [8–11] determine BS/RIS quantities via manual enumeration, which cannot guarantee cost-effective performance and often yield inefficiency optimization.

• **Limitations of evolutionary optimization:** Classical evolutionary algorithms (EAs) are suitable for multi-objective optimization but perform poorly in high-dimensional mixed-integer spaces. They often suffer from premature convergence to local optima and low sampling efficiency when feasible solutions are sparse. Complex joint deployment therefore requires search mechanisms that focus exploration on high-value regions, such as adaptive sampling or surrogate guidance, which are generally absent in standard EA frameworks [12].

Recently, large language models (LLMs) present an entirely new paradigm for addressing RAN planning tasks [13]. They are state-of-the-art pretrained foundation models that combine large-scale pretraining with alignment to support broad generalization and provide feasible solutions for downstream tasks using LLM-based search. Based on a large number of parameters and exceptional learning capabilities [14], their ability to interpret and respond to prompts expressed in natural language, understand intent, and generate solutions opens up new possibilities for solving indoor RAN planning problems in language space, empowered by broad knowledge. To the best of our knowledge, Ref. [15] was the only work that enhances a traditional EA with an LLM, showing the great potential for improving the efficiency of EAs by leveraging LLMs. However, existing research [16,17] for deployment problems has investigated LLM-only approaches, where an LLM serves as the sole optimizer for generating and refining candidate solutions. These approaches achieved optimization efficiency comparable to that of traditional approaches without utilizing the great potential for efficiency improvement. Moreover, carefully designed prompts in these studies are also a key obstacle to automatic prompt engineering, as they require substantial effort to tailor them to specific problems and require expertise in the telecommunications field, resulting in high labor costs [16].

The motivations of this work can be summarized as follows. (1) Simultaneously minimizing the total network cost of BSs and RISs and maximizing communication performance is a bi-objective optimization problem and remains underexplored. (2) Traditional EAs for tackling the bi-objective problem are inefficient due to convergence on local optima, and require redesigning the mechanism to concentrate on the fruitful areas of the search space. (3) Applying

Table 1 Comparison with existing LLM-based deployment optimization studies.

Ref.	Multiple BS/AP deployment	Multiple RIS deployment	Local-optima issue in joint deployment	Dedicated optimization mechanism for solution exploration	Automatic prompt generation
[16]	Yes	No	No	No; LLM-based workflow for modeling, coding, execution, and feedback correction	No; well-crafted and manually designed prompts
[17]	Yes	No	No	Yes; LLM-based optimization framework that uses the LLM as the sole combinatorial optimizer to generate AP-count and AP-placement solutions	No; well-crafted and manually designed prompts
Our paper	Yes	Yes	Yes	Yes; LLM-NSGA-II optimizes the number and position configurations for both BSs and RISs by guiding the search toward promising regions	Yes

LLM-enhanced EAs for improving optimization efficiency remains underexplored in complex joint deployment tasks, and the manual prompt generation for LLMs requires high labor costs to generate a tailored prompt.

1.2 Contributions and organization

Motivated by the above, in this work, we use the LLM-enhanced non-dominated sorting genetic algorithm II (NSGA-II) to directly solve the bi-objective optimization problem, assisted by an LLM to enhance its concentrated exploration capability by extending the exploration in language space. Additionally, we design an LLM-based prompt generation module to automatically generate tailored prompts as an alternative to manual prompt optimization for reducing labor costs. The proposed LLM-enhanced NSGA-II and the prompt generation module together constitute an overall framework, termed LLM-enhanced-EA-driven automated optimization (LEAO). The contributions of this work can be summarized as follows.

- We are the first to employ an LLM-enhanced EA to directly solve the bi-objective optimization problem of joint deployment of BSs and RISs to simultaneously maximize the coverage probability and minimize the network cost.
- We propose LLM-NSGA-II, a novel integration of an LLM into NSGA-II to enhance concentrated exploration by the LLM-suggested solutions. Specifically, these solutions are generated by the LLM in its language space and help guide the exploration towards promising regions in terms of cost-effective solutions in the search space. The mechanism of exploration guidance is achieved by redesigning the traditional operators into intelligent operators. We also have theoretical results regarding the redesign of these new operators.
- We design an LLM-based prompt generation module to automatically generate tailored prompts for the LLM-NSGA-II approach using a prompt template, a user description, reference materials, and prompt examples, eliminating the need for manual prompt optimization to reduce labor costs.
- We validate that the automatic prompt generation module can properly work in the LLM-NSGA-II approach, and compare the LLM-NSGA-II approach with traditional genetic algorithm, NSGA-II, proximal policy optimization (PPO), the LLM-only method, and random deployment in terms of cost efficiency. The results show that our proposed LLM-NSGA-II outperforms other baseline algorithms by showing comparable convergence speed and superior optimal cost efficiency.
- Although developed for the joint deployment of multiple BSs and RISs, the proposed LLM-NSGA-II framework is not limited to wireless network planning. The combination of evolutionary search and LLM-guided exploration is applicable to a broad class of non-convex, mixed-integer, and multi-objective optimization problems, where the search space is combinatorial and conventional evolutionary algorithms suffer from low efficiency or premature convergence. Examples include infrastructure placement, resource allocation, network topology design, and other engineering optimization tasks. Therefore, the proposed LEAO framework provides a general-purpose methodology for LLM-enhanced evolutionary optimization beyond the specific application considered in this work.

Table 1 compares our work with closely related LLM-based deployment optimization studies. Existing studies such as [18,19] mainly focus on BS/AP deployment and rely on manually designed prompts. In this paper, we consider the joint deployment of multiple BSs and multiple RISs. This setting requires the joint selection of BS/RIS numbers and locations. The proposed framework uses an automatic prompt generation module to generate tailored prompts. In this framework, LLM-NSGA-II incorporates a dedicated mechanism for solution exploration that guides the search toward promising regions and helps mitigate premature convergence to local optima.

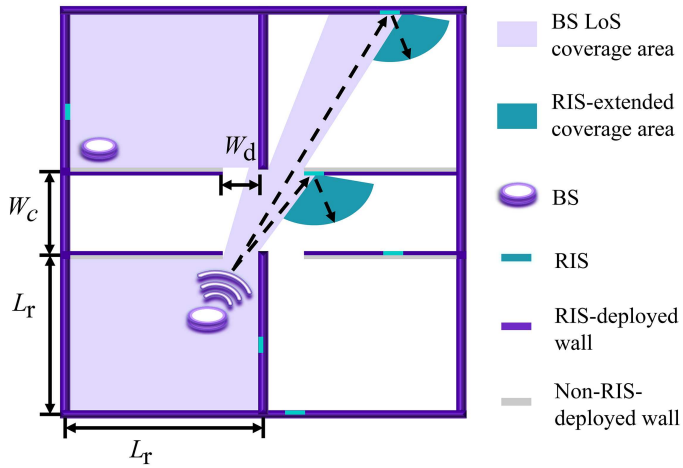

Figure 1 Illustration of the joint indoor deployment with multi-BSs and multi-RISs.

The remainder of this paper is organized as follows. Section 2 formulates the non-convex bi-objective optimization problem for joint indoor deployment. Section 3 describes the LEAO framework architecture. Section 4 presents simulation results and validates our proposed framework, followed by conclusions in Section 5.

2 System model

As shown in Figure 1, we consider that the indoor RAN planning happens in a layout consisting of four square rooms, each with dimensions $L_\mathrm{r} \times L_\mathrm{r}$. The size of a door and the width of the corridor in each of these rooms are indicated by W_d and W_c, respectively. In this indoor environment, BSs are deployed in non-wall areas with their number denoted by N_BS, while N_RIS RISs are not allowed to be deployed on Non-RIS-deployed walls, where placing RISs yields no coverage gain. As a practical deployment constraint, no two RISs can be installed at the same location due to their physical size. Each user equipment (UE) is equipped with a single antenna and is assumed to be randomly distributed within the indoor environment, served by BSs and RISs. Considering a 2-dimensional (2D) coordinate system for the indoor layout, the bottom-left corner of the layout is defined as the origin. Specifically, let (x_i, y_i) and $(\hat{x}_j, \hat{y}_j)$ denote the positions of the i-th BS and the j-th RIS, respectively. Subsequently, both the set of discrete BS positions and the set of discrete RIS positions are given by

$$\mathcal{P}_\mathrm{BS} = \{(x_i, y_i) \mid (x_i, y_i) \in \mathcal{S}_\mathrm{BS}, i \in \mathcal{I}\}, \tag{1}$$

$$\mathcal{P}_\mathrm{RIS} = \{(\hat{x}_j, \hat{y}_j) \mid (\hat{x}_j, \hat{y}_j) \in \mathcal{S}_\mathrm{RIS}, j \in \mathcal{J}\}, \tag{2}$$

where $\mathcal{S}_\mathrm{BS}$ and $\mathcal{S}_\mathrm{RIS}$ are the finite sets of all possible BS and RIS deployment coordinates within the indoor layout, respectively, $\mathcal{I} = \{1, 2, \ldots, N_\mathrm{BS}\}$ and $\mathcal{J} = \{1, 2, \ldots, N_\mathrm{RIS}\}$, $\mathcal{P}_\mathrm{BS} \subseteq \mathcal{S}_\mathrm{BS}$, $\mathcal{P}_\mathrm{RIS} \subseteq \mathcal{S}_\mathrm{RIS}$, and $|\mathcal{P}_\mathrm{BS}| = N_\mathrm{BS}, |\mathcal{P}_\mathrm{RIS}| = N_\mathrm{RIS}$.

2.1 Transmission types and coverage probability

The system mainly considers two types of transmission, i.e., the direct transmission and RIS-assisted transmission. In direct transmission, BSs separately serve the UE without any reflected signals from RISs. Specifically, the LOS transmission is assumed to follow the Rician fading model [18]. For non-line-of-sight (NLOS) transmission, it is assumed to follow the Rayleigh fading model [18] and occurs when transmission is obstructed by walls with a high penetration loss [19–22]. Hence, the received power of direct transmission, denoted as P_r, is

$$P_\mathrm{r} = \begin{cases} \dfrac{P_\mathrm{t}|h|^2\lambda^2}{16\pi^2 d^\alpha}, & n_\mathrm{w} = 0, \\ \dfrac{P_\mathrm{t}\omega^{n_\mathrm{w}}|g|^2\lambda^2}{16\pi^2 d^\alpha}, & \text{otherwise}, \end{cases} \tag{3}$$

where λ and α are the signal wavelength and pathloss exponent, respectively, P_t, d, and ω denote the transmit power, transmission distance, and wall penetration loss, respectively, n_w denotes the number of penetration walls, h and g follow the Rician and Rayleigh distributions, respectively.

Alternatively, RIS-assisted transmission works if two conditions are met: (1) the direct transmission from BSs to the UE is blocked by walls; (2) available LoS transmission exists in both the BS-to-RIS and RIS-to-UE transmission. Specifically, the signal is initially transmitted from the BS to the RIS and is then reflected towards the UE. Following that, orthogonal sub-bands are considered to be allocated to BSs to avoid shared frequency conflicts, taking advantage of the extensive bandwidth of mmWave networks [23]. Each signal undergoes only a single reflection. Considering the physical properties of RIS, each RIS is made of M unit cells, where $M = \nu \times \hat{\nu}$, ν and $\hat{\nu}$ indicate the numbers of rows and columns of unit cells, respectively. For each RIS unit cell, the physical size and gain are indicated by $(d_{\mathrm{u}})^2$ and G, respectively. Although the phase shift is different, all unit cells have the same amplitude value A of the reflection coefficient. We further assume that the power radiation pattern of RISs is independent of the incident angle and the reflected angle of the signal. For received signal power enhancement, the reflected signals of unit cells are aligned in phase [24] with perfect phase shift control. Consequently, the peak radiation is achieved in all directions. Hence, the received signal power of the UE via RIS-assisted transmission [25] is given by

$$P_{\mathrm{r}} = \frac{P_{\mathrm{t}} G \left(\sum_{m=1}^{M} \phi_m \varphi_m\right)^2 d_{\mathrm{u}}^2 \lambda^2 A^2}{64\pi^3 d_1^2 d_2^2}, \tag{4}$$

where ϕ_m is the fading coefficient for the link between the BS and the m-th unit cell of the RIS, φ_m denotes the link between the m-th unit cell and the UE, wherein ϕ_m and φ_m follow Rician distribution, d_1 and d_2 are the distance between the UE and the RIS and the distance between the BS and the RIS, respectively.

After the indoor BSs-and-RISs joint deployment, the coverage performance of a deployment solution is calculated by a Monte-Carlo simulation approach for N_{g} simulation times. Each simulation randomly generates the location of the UE in the indoor environment [26]. Based on the joint deployment and UE position, the received signal power of the UE is calculated. Furthermore, the UE is considered to be covered if its received signal power exceeds T_{r}, where T_{r} denotes the received-signal-power threshold (RSPT). Therefore, the coverage probability is defined as the probability that the received signal power exceeds a given threshold, i.e.,

$$C \triangleq \Pr(P_{\mathrm{r}} \geqslant T_{\mathrm{r}}) = \mathbb{E}[\mathbf{1}\{P_{\mathrm{r}} \geqslant T_{\mathrm{r}}\}]. \tag{5}$$

In the Monte-Carlo simulations with N_{g} independent trials, C is estimated by the sample mean

$$\widehat{C} = \frac{1}{N_{\mathrm{g}}} \sum_{k=1}^{N_{\mathrm{g}}} \mathbf{1}\left\{P_{\mathrm{r}}^{(k)} \geqslant T_{\mathrm{r}}\right\}, \tag{6}$$

where $P_{\mathrm{r}}^{(k)}$ denotes the received signal power obtained in the k-th trial.

2.2 Problem formulation

Recall that both indoor coverage and network cost vary in deployment solutions, and these two metrics are primary concerns in indoor RAN planning. Accordingly, instead of formulating two separate single-objective problems, we recast the task as a bi-objective optimization problem that simultaneously minimizes the network cost η and maximizes the coverage probability C by optimizing the number and deployment positions of both BSs and RISs. The network cost η of a deployment solution is given by

$$\eta = N_{\mathrm{BS}} \cdot \eta_{\mathrm{BS}} + N_{\mathrm{RIS}} \cdot \eta_{\mathrm{RIS}}, \tag{7}$$

where η_{BS} and η_{RIS} are the unit costs of BSs and RISs, respectively. These unit costs account for three factors: (1) the cost of each piece of equipment, (2) the length of the installment plan, and (3) the interest rate [27, 28]. Then, the highest number of BSs and RISs can be expressed as $N_{\mathrm{BS}}^{\mathrm{H}}$ and $N_{\mathrm{RIS}}^{\mathrm{H}}$, respectively. Hence, the set of BS positions with $N_{\mathrm{BS}}^{\mathrm{H}}$ and the set of RIS positions with $N_{\mathrm{RIS}}^{\mathrm{H}}$ can be expressed as

$$\mathcal{P}'_{\mathrm{BS}} = \{(x_i, y_i) \mid (x_i, y_i) \in \mathcal{S}_{\mathrm{BS}},\, i \in \mathcal{I}'\}, \tag{8}$$

$$\mathcal{P}'_{\mathrm{RIS}} = \{(\hat{x}_j, \hat{y}_j) \mid (\hat{x}_j, \hat{y}_j) \in \mathcal{S}_{\mathrm{RIS}},\, j \in \mathcal{J}'\}, \tag{9}$$

where $\mathcal{I}' = \{1, 2, \ldots, N_{\mathrm{BS}}^{\mathrm{H}}\}$ and $\mathcal{J}' = \{1, 2, \ldots, N_{\mathrm{RIS}}^{\mathrm{H}}\}$.

Given a deployment solution $\mathbf{x} = (N_{\mathrm{BS}}, N_{\mathrm{RIS}}, \mathcal{P}_{\mathrm{BS}}, \mathcal{P}_{\mathrm{RIS}})$, the system model and Monte-Carlo evaluation yield the corresponding coverage probability and network cost, denoted by $C(\mathbf{x})$ and $\eta(\mathbf{x})$, respectively. Hence, $C : \mathcal{X} \to [0, 1]$

and $\eta : \mathcal{X} \to \mathbb{R}_+$, with $\mathbf{x} \mapsto C(\mathbf{x})$ and $\mathbf{x} \mapsto \eta(\mathbf{x})$, are performance mappings from the feasible deployment set $\mathcal{X}$ to their metric values. The joint optimization is then expressed in a bi-objective problem **BP** as

$$\mathbf{BP}: \quad \min_{\mathbf{x}} \left[\eta(\mathbf{x}),\, 1 - C(\mathbf{x}) \right], \tag{10a}$$

$$\text{s.t. } C(\mathbf{x}) \geqslant C_{\min}, \tag{10b}$$

$$\eta_{\min} \leqslant \eta(\mathbf{x}) \leqslant \eta_{\max}, \tag{10c}$$

$$N_{\mathrm{BS}} \in \left\{1, \ldots, N_{\mathrm{BS}}^{\mathrm{H}}\right\}, \tag{10d}$$

$$N_{\mathrm{RIS}} \in \left\{0, \ldots, N_{\mathrm{RIS}}^{\mathrm{H}}\right\}, \tag{10e}$$

$$\mathcal{P}_{\mathrm{BS}} \subseteq \mathcal{S}_{\mathrm{BS}}, \tag{10f}$$

$$\mathcal{P}_{\mathrm{RIS}} \subseteq \mathcal{S}_{\mathrm{RIS}}, \tag{10g}$$

where $C_{\min}$ is the minimum coverage probability required for an available deployment solution, and $(\eta_{\min}, \eta_{\max})$ bounds the allowable cost range imposed by budgetary constraints or practical deployment limits. To clarify, passive RIS cannot be deployed without BSs and thus constrains the lower bound on cost to ensure at least one BS is deployable within the budget. As a result, this optimization problem aims to maximize coverage probability C while minimizing network cost η.

Problem **BP** possesses the non-convex, NP-hard property towards balancing the trade-off between indoor coverage probability and network cost [5,9], owing to (i) a tremendous amount of placement options for $\mathcal{P}_{\mathrm{BS}}$ and $\mathcal{P}_{\mathrm{RIS}}$ across a 2D indoor layout constrained by room geometry, (ii) balancing this two-factor trade-offs involves a combinatorial optimization problem for N_{BS} and N_{RIS}, and (iii) solving this bi-objective optimization problem involves a nonlinear programming process where the bi-objective function shown in (10) is not a linear function and the calculation of the minima $\eta(\mathbf{x})$ and $(1 - C(\mathbf{x}))$ is over a set of unknown real variables and conditional to the constraints, i.e., Eqs. (10b)–(10g).

3 LLM-enhanced-EA-driven automated optimization framework for joint indoor deployment

To enhance the optimization efficiency of solving the optimization problem **BP**, we propose an LLM-enhanced EA method and further propose an LLM-enhanced-EA-driven automated optimization framework. To efficiently search for effective solutions, we design intelligent operators in the LLM-NSGA-II to leverage LLM-suggested solutions for guided search. This LLM-enhanced EA approach differs from the classic methods in the following aspects.

• Avoid LLM-only optimization: In this paper, the LLM operates as a co-optimizer alongside an EA rather than serving as the sole optimizer for generating and refining candidate solutions. In [16, 17], optimization relies exclusively on LLMs to produce improved solutions. Although pre-trained LLMs possess broad knowledge, they lack specialized expertise in specific domains [13], and their performance varies significantly when applied to optimization tasks outside their design scope [29]. Consequently, LLM-only methods often underperform and fail to achieve high best-of-run objective values across iterations compared with EAs.

• Avoid EA-only optimization: The proposed approach combines an EA (NSGA-II) with an LLM-based co-optimizer instead of relying solely on EAs. As reported in [8,9,30], EAs are widely used for non-convex optimization problems in RAN planning. While they can yield near-optimal solutions [29, 31], they often converge slowly. In this work, the hybrid approach can maintain the near-optimality of EAs and take advantage of the LLM for faster convergence. We employ the fast, elitist non-dominated sorting genetic algorithm (NSGA-II) as the EA component because it is widely applied for bi-objective optimization and remains influential in engineering [30].

• Avoid labor-intensive prompt customization: In this work, the proposed prompt generation module assembles a task-specific prompt to address the key challenge of labor-intensive manual human intervention in prompt engineering. Human intervention is pervasive in prompt engineering, particularly for crafting prompts tailored to specific tasks. However, such intervention demands sustained, labor-intensive effort and task-specific expertise for joint deployment scenarios. In this work, we investigate automatic prompt generation as an alternative to the classic manual approach, mitigating labor-intensive human intervention.

These gaps remain underexplored in real-world telecommunications engineering scenarios and substantially increase the difficulty of solving the optimization problem **BP** when relying solely on LLM or EA approaches [8, 9, 16, 17]. To overcome these obstacles, we introduce a novel LEAO framework. The details of its implementation are presented in the following subsections.

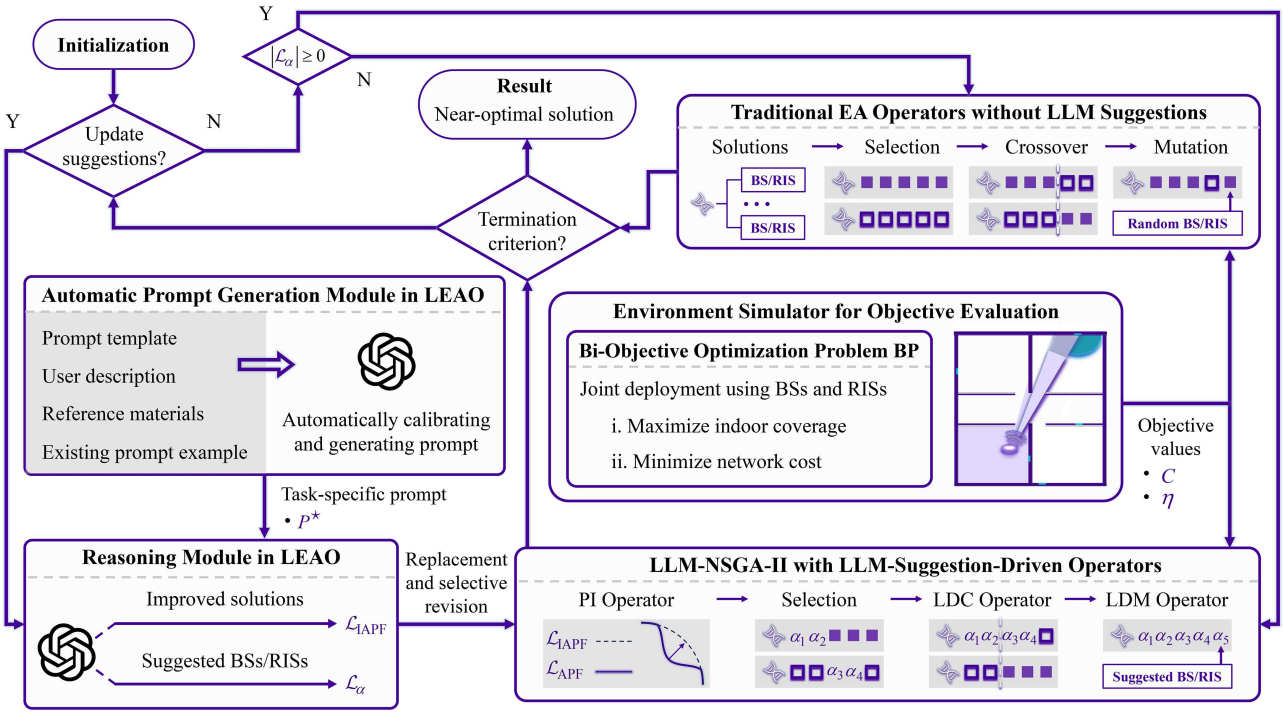

Figure 2 An illustration of the LEAO framework that shows the distinction between EAs using traditional genetic operators and LLM-NSGA-II using LLM-suggestion-driven operators.

3.1 Overall architecture

As shown in Figure 2, the proposed LEAO framework has two components: (i) an LLM-NSGA-II that uses LLM guidance to generate and refine candidate solutions, thereby steering the EA toward LLM-suggested regions of the potential solution space for efficient optimization; and (ii) an in-context prompt generation module (PGM) that assembles task-specific prompts from a template, a user description, prompt examples, and reference materials for the joint deployment task, mitigating labor-intensive human intervention in prompt engineering. Specifically, LEAO first initiates an EA search with a set of randomly generated candidate solutions $\mathbf{x}$ as the initial population (as in the standard NSGA-II algorithm) for the joint deployment problem **BP**. Then, these candidate solutions are evaluated through an environment simulator (ES) to calculate their performance in terms of coverage probability C and network cost η. Next, based on the given materials, PGM assembles a prompt tasked with optimizing the evaluated solutions, provides it to the reasoning module (RM), and uses an LLM to solve the task, resulting in a set of improved solutions and partial solutions advantageous to solving the problem. Subsequently, novel intelligent operators in LLM-NSGA-II guide the EA search to focus on LLM-generated solutions in two ways. Intelligent operators replace some of the candidate solutions with improved solutions that possess superior objective values. Meanwhile, they assemble new solutions by integrating original candidates with partial solutions from the LLM. As a whole, these form an LLM-suggestion-driven NSGA-II system for solving the joint deployment problem by guiding the solution search toward promising regions of the potential solution space. In each iteration (termed a generation in EAs), the optimization cycle repeats, and finally, the evolutionary loop ends when a predefined stopping criterion is met.

3.2 LLM-suggestion-driven non-dominated sorting genetic algorithm II

In this subsection, we explain how the LLM-suggestion-driven operators generate new variations of candidate solutions $\mathbf{x}$ and integrate LLM-suggested solutions into the iterative optimization process of LLM-NSGA-II. Unlike traditional genetic operators (such as standard crossover and mutation) [30], these intelligent operators introduce two novel mechanisms: (i) the positive selection of partial solutions advantageous to the deployment objectives, and (ii) the replacement of certain candidate solutions with those proposed by the LLM. Based on these mechanisms, the intelligent operators guide the EA to focus on promising regions of the solution space. We then summarize our approach that leverages LLM-based search to augment NSGA-II with the theoretical support from both evolutionary biology and economics, and clarify in the context of indoor RAN planning.

In order to jointly optimize both the placement and numbers of BSs and RISs, we encode each candidate deployment solution $\mathbf{x}$ as a unified vector $c = [N_{\mathrm{BS}}, N_{\mathrm{RIS}}, \mathcal{P}'_{\mathrm{BS}}, \mathcal{P}'_{\mathrm{RIS}}]$. Here, $\mathcal{P}'_{\mathrm{BS}}$ and $\mathcal{P}'_{\mathrm{RIS}}$ are ordered lists of potential BS coordinates and RIS coordinates, respectively. A specific deployment configuration $\mathbf{x}$ is obtained by taking the first N_{BS} entries of $\mathcal{P}'_{\mathrm{BS}}$ as the active BS locations and the first N_{RIS} entries of $\mathcal{P}'_{\mathrm{RIS}}$ as the active RIS locations. Thus, modifying elements of the unified vector c yields variations in the candidate solution, allowing exploration of different indoor deployment plans. We assume that the LLM provides two forms of guidance: (i) complete candidate solutions, and (ii) partial solutions in the form of specific BS or RIS placements that it identifies as beneficial for achieving high coverage probability C and low cost η. The LLM also specifies the positions of these beneficial BS/RIS placements within the unified vector c.

Before integrating LLM suggestions to enhance traditional operators, we adopt the fundamental EA operators as follows.

• *Mutation:* This operator makes a small random change to one or more decision variables of a single candidate solution to generate a new solution in its neighborhood. In EA terminology, each decision variable (e.g., N_{BS}, N_{RIS}, or a coordinate in $\mathcal{P}'_{\mathrm{BS}}/\mathcal{P}'_{\mathrm{RIS}}$) is referred to as a gene/allele, and a full set of decision variables defining one solution is referred to as a chromosome [30]. In our context, treating N_{BS}, N_{RIS}, and each coordinate in $\mathcal{P}'_{\mathrm{BS}}$ and $\mathcal{P}'_{\mathrm{RIS}}$ as genes means that mutating a gene corresponds to changing either the number of BSs/RISs or altering one specific BS or RIS location in the deployment solution.

• *Crossover:* This operator recombines two parent solutions to form one or more offspring solutions. It typically involves exchanging subsets of genes between two selected parent chromosomes. For example, in a single-point crossover, a random cut point is chosen in the chromosome, and the genes after that point are swapped between the two parents to produce new offspring. In our context, a crossover takes two indoor deployment plans (termed parent chromosomes) and mixes their BS and RIS configuration segments, yielding new candidate deployments (offspring chromosomes).

3.2.1 *LLM-driven mutation (LDM)*

The LLM-driven mutation operator leverages LLM guidance during mutation. Instead of purely random mutations, LDM can replace one or more BS/RIS positions in a candidate solution with alternative positions suggested by the LLM. In this way, the mutation is biased toward regions of the search space that the LLM expects to be promising for indoor coverage and cost. In this work, compared to a standard mutation that might change a BS/RIS position to a random new value, LDM preferentially mutates genes to known beneficial BS/RIS coordinates recommended by the LLM.

We formally incorporate this bias to design the LLM-driven mutation operator. Each LLM-suggested beneficial coordinate is assumed as an advantageous allele at its corresponding gene index (termed locus) in the chromosome. First, the LDM operator performs a sequence mapping using the indices of these advantageous alleles to identify their locations on the chromosome c. Next, LDM divides the mutation probability for any selected gene between two possibilities: mutating it to one of its advantageous allele values, or randomly mutating it to any other value. A hyperparameter $\beta_{\mathrm{M}} \in (0, 1)$ controls this probability split. By default, a fraction β_{M} of the mutation probability is allocated to switching the gene to an advantageous allele, and the remaining $1 - \beta_{\mathrm{M}}$ is reserved for mutating to any other value. If a target gene has multiple advantageous allele options, e.g., several promising coordinates identified for a particular BS, the β_{M} probability mass is evenly divided among those options. This mechanism increases selection pressure toward LLM-suggested beneficial genes while still preserving some randomness for exploration. It allows practitioners to balance favoring advantageous LLM hints against maintaining genetic diversity in the population. Let $\mathcal{L}_{\mathbf{x}} = \{0, 1, \ldots, 1 + N_{\mathrm{BS}}, \ldots, 1 + N_{\mathrm{BS}} + N_{\mathrm{RIS}}\}$ and $\mathcal{L}_c = \{0, 1, \ldots, 1 + N^{\mathrm{H}}_{\mathrm{BS}}, \ldots, 1 + N^{\mathrm{H}}_{\mathrm{BS}} + N^{\mathrm{H}}_{\mathrm{RIS}}\}$ denote the sets of indices that contain advantageous alleles for the specific solution $\mathbf{x}$ and for the chromosome c, respectively. The probability that a gene selected for mutation will mutate into one of its advantageous alleles is then derived as

$$\begin{aligned} P^{\mathrm{M}'}_k &= \mathbb{P}\left(e'_k \in \{\alpha_{k,1}, \ldots, \alpha_{k,A_k}\} \,\middle|\, L_k \in \mathcal{L}_c,\, S^{\mathrm{M}}_k = 1\right) \\ &= \sum_{a=1}^{A_k} \mathbb{P}\left(e'_k = \alpha_{k,a} \,\middle|\, L_k \in \mathcal{L}_c,\, S^{\mathrm{M}}_k = 1\right) \\ &= \sum_{a=1}^{A_k} \frac{\beta_{\mathrm{M}}}{A_k}, \end{aligned} \tag{11}$$

where e_k is the k-th gene in c and e'_k is its value after mutation, $\alpha_{k,a}$ is the a-th advantageous allele available to

gene e_k, A_k is the total number of advantageous alleles identified for e_k, L_k is the locus of e_k, and $S_k^{\mathrm{M}} = 1$ indicates that the gene has been selected for mutation. Given an overall mutation probability $p_{\mathrm{M}} \in (0,1)$ for each gene, the probability that a gene with at least one advantageous allele will mutate into one of those advantageous alleles is

$$\begin{aligned} P_k^{\mathrm{M}} &= \mathbb{P}\left(e'_k \in \{\alpha_{k,1}, \ldots, \alpha_{k,A_k}\},\, S_k^{\mathrm{M}} = 1 \,\middle|\, L_k \in \mathcal{L}_c\right) \\ &= p_{\mathrm{M}} P_k^{\mathrm{M}'} = p_{\mathrm{M}} \sum_{a=1}^{A_k} \frac{\beta_{\mathrm{M}}}{A_k}. \end{aligned} \tag{12}$$

Based on this formulation, the LDM operator biases mutations to preferentially preserve LLM-suggested BSs/RISs for joint indoor deployment. It focuses the search on candidate solutions that are especially relevant according to the LLM's broad knowledge, for instance, indoor deployment configurations likely to improve coverage and cost. At the same time, by assigning $1-\beta_{\mathrm{M}}$ of mutation probability to other random variations, LDM maintains exploratory behavior to avoid premature convergence. In summary, LDM provides a controllable balance: it can intensify the retention of LLM-suggested beneficial BSs/RISs while still ensuring a diverse set of solutions is explored.

3.2.2 *LLM-driven crossover (LDC)*

The LDC operator similarly uses LLM insights to bias the recombination process in favor of advantageous genes. Consider a single-point crossover example with two deployment solutions treated as parent chromosomes c_{p} and c_{q}. Traditional single-point crossover randomly chooses a crossover point along the length of c for generating offspring solutions. In LDC, however, we modify this by increasing the likelihood of crossover points that result in offspring retaining more of the LLM-suggested advantageous alleles from the parents.

LDC first maps out all advantageous allele indices present in either parent. Let $\mathcal{A}_{\mathrm{p}}$ and $\mathcal{A}_{\mathrm{q}}$ be the sets of indices of advantageous alleles on parent c_{p} and c_{q}, respectively, and $\mathcal{A}_{\mathrm{pq}} = \mathcal{A}_{\mathrm{p}} \cup \mathcal{A}_{\mathrm{q}}$ be the union of these indices. Among all possible crossover points, there will be certain points at which cutting the chromosomes and swapping tails yields offspring solutions that carry the maximum possible number of advantageous alleles from $\mathcal{A}_{\mathrm{pq}}$. Define the set of such crossover points as $\mathcal{A}'_{\mathrm{pq}} = \{\ell_{\mathrm{pq},1}, \ldots, \ell_{\mathrm{pq},R_{\mathrm{pq}}}\}$, where $R_{\mathrm{pq}} = |\mathcal{A}'_{\mathrm{pq}}|$. LDC assigns a higher probability to selecting a crossover point from $\mathcal{A}'_{\mathrm{pq}}$ than other points. Specifically, if a parent pair is selected for crossover and this event is indicated by $S_{\mathrm{pq}}^{\mathrm{C}} = 1$, we define a hyperparameter $\beta_{\mathrm{C}} \in (0,1)$ such that a fraction β_{C} of the conditional probability is dedicated to points in $\mathcal{A}'_{\mathrm{pq}}$. Accordingly, the probability of choosing a crossover point ℓ'_{pq} that lies in the set of points maximizing advantageous allele retention is

$$\begin{aligned} P_{\mathrm{pq}}^{\mathrm{C}'} &= \mathbb{P}\Big(\ell'_{\mathrm{pq}} \in \{\ell_{\mathrm{pq},1}, \ldots, \ell_{\mathrm{pq},R_{\mathrm{pq}}}\} \,\Big|\, |\mathcal{A}_{\mathrm{pq}}| > 0,\, S_{\mathrm{pq}}^{\mathrm{C}} = 1\Big) \\ &= \sum_{r=1}^{R_{\mathrm{pq}}} \mathbb{P}\Big(\ell'_{\mathrm{pq}} = \ell_{\mathrm{pq},r} \,\Big|\, |\mathcal{A}_{\mathrm{pq}}| > 0,\, S_{\mathrm{pq}}^{\mathrm{C}} = 1\Big) \\ &= \sum_{r=1}^{R_{\mathrm{pq}}} \frac{\beta_{\mathrm{C}}}{R_{\mathrm{pq}}}, \end{aligned} \tag{13}$$

where we condition on $|\mathcal{A}_{\mathrm{pq}}| > 0$, indicating there is at least one advantageous allele between the two parents. Based on the above construction, $\mathcal{A}'_{\mathrm{pq}} \subseteq \mathcal{A}_{\mathrm{pq}}$, and if $|\mathcal{A}_{\mathrm{pq}}| = 0$ (i.e., neither parent carries any advantageous allele), LDC defaults to a standard random crossover. Once $S_{\mathrm{pq}}^{\mathrm{C}} = 1$, indicating the pair is selected for crossover, the probability that this pair undergoes crossover at one of the points in $\mathcal{A}'_{\mathrm{pq}}$ is given by

$$\begin{aligned} P_{\mathrm{pq}}^{\mathrm{C}} &= \mathbb{P}\Big(\ell'_{\mathrm{pq}} \in \{\ell_{\mathrm{pq},1}, \ldots, \ell_{\mathrm{pq},R_{\mathrm{pq}}}\},\, S_{\mathrm{pq}}^{\mathrm{C}} = 1 \,\Big|\, |\mathcal{A}_{\mathrm{pq}}| > 0\Big) \\ &= p_{\mathrm{C}} P_{\mathrm{pq}}^{\mathrm{C}'} = p_{\mathrm{C}} \sum_{r=1}^{R_{\mathrm{pq}}} \frac{\beta_{\mathrm{C}}}{R_{\mathrm{pq}}}, \end{aligned} \tag{14}$$

where $p_{\mathrm{C}} \in (0,1)$ is the overall crossover probability for any selected pair. As a result, LDC biases the crossover operation so that if advantageous BSs/RISs are present, the resulting offspring are more likely to inherit them. Therefore, this biased retention of LLM-suggested beneficial BSs/RISs ensures that favorable partial solutions are propagated in subsequent generations, while $1-\beta_{\mathrm{C}}$ of the crossover selection probability still accounts for ordinary crossover points to maintain diversity during solution exploration.

3.2.3 *Pareto improvement operator (PI)*

The PI operator is designed to inject high-quality LLM-proposed solutions into the EA population without disrupting the optimality of existing solutions. To this end, we combine the current approximate Pareto front (APF) [30] of the population with an improved set of solutions from the LLM, and then filter to retain only the best non-dominated solutions of the combined set as the new APF.

First, to form the improved set, we use an integer hyperparameter β_{I} to select the β_{I} most superior candidate solutions in terms of their multi-objective performance from the current APF. Then, the reasoning module of our framework queries the LLM with these β_{I} solutions, obtaining a one-to-one set of LLM-refined solutions. This yields an improved approximate Pareto front (IAPF) of size β_{I}. Next, we form the union $\mathcal{L}_{\mathrm{U}} = \mathcal{L}_{\mathrm{APF}} \cup \mathcal{L}_{\mathrm{IAPF}}$, which can contain at most $|\mathcal{L}_{\mathrm{APF}}| + |\mathcal{L}_{\mathrm{IAPF}}|$ solutions as double the size of the original APF. The PI operator evaluates all solutions in $\mathcal{L}_{\mathrm{U}}$ and then performs an extra round of non-dominated sorting and crowding-distance assignment on this combined set. After sorting, we select a subset of size β_{P} from $\mathcal{L}_{\mathrm{U}}$ using the standard crowded tournament selection procedure [32], which is commonly used to enforce a limit on Pareto front size while preserving solution diversity and maintaining nondomination of the resulting subset. The value of β_{P} denotes the number of solutions to retain and is chosen such that $\beta_{\mathrm{P}} \leqslant |\mathcal{L}_{\mathrm{APF}}|$. Typically, we set $\beta_{\mathrm{P}} = |\mathcal{L}_{\mathrm{APF}}|$ by default to maintain the original Pareto front size. Finally, the nondomination rank of every chromosome in the retained set is reset to zero before continuing to the next generation, ensuring that the new APF solutions are treated as top-tier solutions in the population.

Based on the above construction, choosing $\beta_{\mathrm{P}} \leqslant |\mathcal{L}_{\mathrm{APF}}|$ guarantees that none of the solutions retained from $\mathcal{L}_{\mathrm{U}}$ are dominated by any solution that was in the original APF. To prove that, let $\overline{\mathcal{L}_{\mathrm{APF}}}$ denote the set of solutions in $\mathcal{L}_{\mathrm{U}}$ that were not part of the original APF. We assume that within $\mathcal{L}_{\mathrm{U}}$, all solutions have been correctly assigned a nondomination rank after sorting. For any original APF solution $c \in \mathcal{L}_{\mathrm{APF}}$ and any new solution $\overline{c} \in \overline{\mathcal{L}_{\mathrm{APF}}}$, by definition, c is not dominated by $\overline{c}$ (since c belongs to the APF) and each $\overline{c}$ is dominated by at least one c. If an LLM-proposed solution $c' \in \mathcal{L}_{\mathrm{IAPF}}$ happened to be dominated by some $\overline{c}$, then because c dominates $\overline{c}$ and $\overline{c}$ dominates c', it follows that c would dominate c' as well. Thus, as long as we limit β_{P} to $|\mathcal{L}_{\mathrm{APF}}|$, the crowded tournament selection on $\mathcal{L}_{\mathrm{U}}$ will ensure that only solutions at least as good as the original APF members are retained, thereby preserving the non-dominated property of the Pareto front. As a result, this novel operator preserves (1) the number of solutions on the Pareto front and (2) the population size across generations. Therefore, the PI operator injects LLM-proposed solutions in a controlled way that maintains Pareto optimality and diversity, strengthening the EA's search with high-quality suggestions without disrupting the existing good solutions.

It is worth noting that LDM, LDC and PI are inspired by notions from evolutionary biology and economics. On the one hand, our design of LDM and LDC is analogous to mechanisms observed in evolutionary biology, which provides additional intuition for their effectiveness. Specifically, how LDM and LDC favor beneficial BSs/RISs for cost-effective joint indoor deployment can be likened to the concept of positive selection of advantageous genetic traits. For instance, the antagonistic pleiotropy hypothesis proposed by Williams in 1957 suggests that certain gene mutations are retained in a population because they confer early-life advantages, even if they have adverse effects later in life [33]. Building on this idea, recent genome-wide evidence by Long et al. [34] showed that alleles beneficial to reproduction tend to be selectively favored in humans. In standard EAs (e.g., GA and NSGA-II), however, such directed preservation of beneficial gene values is not explicitly implemented. By introducing LDM and LDC, which actively preserve and propagate LLM-suggested "advantageous alleles" (beneficial BS/RIS positions) during mutation and crossover, our framework explicitly incorporates this principle of positive selection. Figure 2 illustrates how LDM differs from a traditional mutation operator and how LDC differs from a traditional crossover operator. On the other hand, our PI operator is grounded in the principle of Pareto optimality, a concept from economics. In neoclassical economics, a Pareto improvement is a change that makes at least one individual better off without making anyone worse off [35]. We applied this principle to the evolutionary search by ensuring that when LLM-generated solutions are added to the population, they do not cause any deterioration in the optimization objectives of existing solutions. Thus, the PI operator guarantees that the population is improved toward better Pareto fronts using LLM without sacrificing any previous gains, directly reflecting the Pareto improvement philosophy in the optimization process.

In summary, the proposed intelligent operators (i.e., LDM, LDC, and PI) work in concert to guide the EA's search toward regions of the solution space that the LLM deems promising. LDM and LDC operate at the variable level to inject LLM-informed variations (favoring advantageous BS/RIS placements), while PI operates at the solution level to incorporate whole LLM-suggested solutions in the manner of maintaining Pareto optimality. By combining the exploratory power of EAs with the domain-informed guidance of LLMs, this LLM-enhanced NSGA-II can enhance the optimization efficiency for exploring high-quality joint indoor deployment solutions in terms of higher coverage

and lower cost, as shown in Figure 2.

3.3 Automatic prompt generation module

This subsection proposes an automatic prompt generation module that automates the creation of prompts using four key elements: (1) a prompt template, (2) reference materials, (3) a user description, and (4) an existing prompt example, as shown in Figure 3. Based on the in-context learning approach [36], the module combines these elements, prompting the LLM to generate a custom, task-specific prompt for solving the problem **BP**. The primary benefit is reduced manual intervention when creating a prompt using this module.

3.3.1 *Prompt template*

This element provides a high-level overview of the optimization problem. While specific tasks may require different prompt designs [17, 37], the general prompt template focuses on what is common across optimization tasks rather than task-specific details. It is structured around four components: role, inputs, workflow, and output requirements, following a general design framework [36]. Certain components include broad "hints" as constraint phrases in a generalized form, reducing the risk of LLM hallucination. Other components, such as concrete inputs and explicit output requirements, are manually specified to ensure stable performance. This general template suits users without in-depth domain expertise in joint indoor deployment, significantly reducing the need for labor-intensive prompt customization.

3.3.2 *Reference materials*

This LLM-powered element automatically extracts, classifies, and filters key information from the source code of the environment simulator. LLMs can derive high-level concepts from detailed content [38] and excel at code summarization [39]. By leveraging these capabilities, the element distills the long-form code into key entities and relationships. It then produces LLM-generated supplementary documentation that consolidates parameter definitions, constraints, and representative examples. All extracted content is traceable back to the environment simulator, resulting in grounded, task-specific materials.

3.3.3 *User description*

This element is a set of natural-language instructions that guide the LLM on how to use the content from the four elements to form the final prompt. Based on these instructions, the LLM is prompted to generate a thorough prompt tailored to the specific optimization task for joint indoor deployment. To mitigate the need for human intervention, the instructions are crafted to be understandable without requiring deep domain expertise in joint indoor deployment. As a result, this element reduces the need for intensive manual effort, making the prompt construction process labor-saving and accessible to general users.

3.3.4 *Existing prompt example*

This element provides one or more complete prompt examples taken from existing published papers, serving as few-shot examples [36] for the construction of the final prompt. By presenting the LLM with examples of well-crafted, manually designed prompts, this element guides it to produce a task-specific prompt that emulates these high-quality examples.

By combining these four elements, the module gathers all the necessary content needed to instruct the LLM and finally yields a task-specific prompt tailored to problem **BP**. This in-context learning approach relies primarily on automated, labor-saving components with minimal human input. The resulting prompt follows the structure of the general template and is enriched with details from the automated environment explanation. As a result, the automatic prompt generation module alleviates the need for labor-intensive prompt engineering. It also improves the automation and practicality of our approach in real-world applications.

3.4 Reasoning module

The reasoning module comprises three functions: (1) prompt variable initialization, which inserts actual variable values into the prompt by replacing placeholders without changing the prompt's structure; (2) LLM prompting, which uses the prompt to invoke the LLM and generate optimized results; and (3) structured output, which configures the LLM to return results in the structured format required by LLM-NSGA-II.

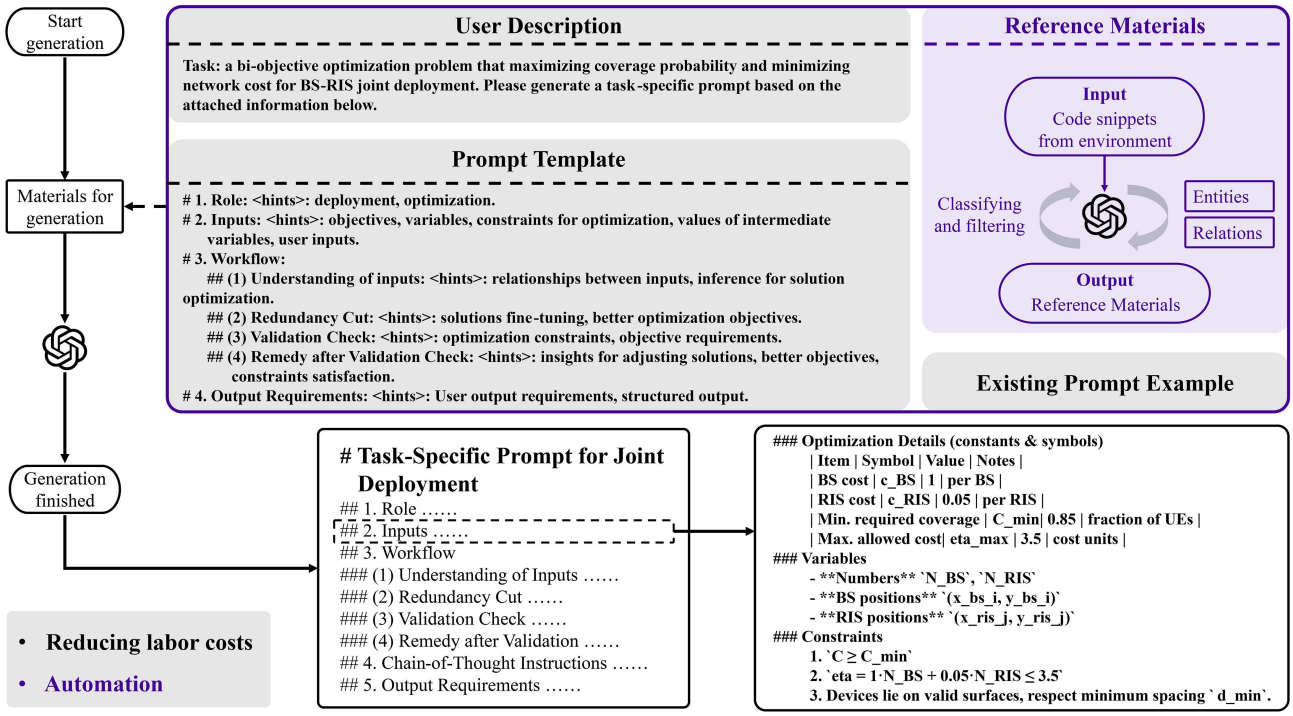

Figure 3 Task-specific prompt generated by the automatic prompt generation module.

3.4.1 *Prompt variable initialization*

To set up the task-specific prompt, this function replaces all placeholders in the prompt inputs with actual data before calling the LLM. In prompt engineering, placeholders are filled at runtime with dynamic content[3)]. For example, this function automatically inserts the current APF, which contains the unimproved deployment solutions, directly into the prompt at runtime.

3.4.2 *LLM prompting*

This function sends the task-specific prompt to an LLM via an API call[3)] to obtain optimized results. Based on the prompt, the LLM follows instructions from the automatic prompt generation module to address problem **BP**. Finally, its response includes the improved approximate Pareto front (IAPF), representing a set of solutions optimized by the LLM, as well as any identified advantageous alleles required by LLM-NSGA-II.

3.4.3 *Structured output*

This function ensures that the LLM's output is returned in the required format by using a structured output schema. Many LLM APIs support specifying a desired output format[3)], which prevents the model from omitting required information. With a structured output, the LLM directly provides the IAPF and advantageous alleles in a consistent format to LLM-NSGA-II, eliminating the need for manual reformatting.

As a result, the reasoning module outputs the generated IAPF and advantageous alleles to the intelligent operators in LLM-NSGA-II.

3.5 Complexity analysis

We analyze the computational complexity under the standard NSGA-II framework. LEAO framework is summarized in Algorithm 1. The standard NSGA-II operations, including fast non-dominated sorting and crowding distance assignment, are implemented according to [40].

3.5.1 *LLM-NSGA-II*

In both NSGA-II and LLM-NSGA-II, the per-generation time complexity (excluding the LLM inference overhead) is dominated by the most expensive steps in the evolutionary loop. Let N and J denote the population size and the

3) OpenAI. The OpenAI API. 2025. Accessed: 04-Oct-2025.

Algorithm 1 The LEAO framework for **BP**.

Input: Initial parameters of the LEAO;
Initialization:
$\mathcal{P} \leftarrow$ CreateInitialPopulation();
Call ES($\mathcal{P}$) to evaluate bi-objective values $\{C, \eta\}$ of every chromosome in $\mathcal{P}$;
Call FastNon-DominatedSorting($\mathcal{P}$) and Crowding-DistanceAssignment($\mathcal{P}$) to evaluate $\mathcal{P}$, and get $\mathcal{L}_{\mathrm{APF}}$;
for $n_{\mathrm{g}} = 1, 2, \cdots, N_{\mathrm{g}}$ **do**
  Empty $\mathcal{O}$;
  if $n_{\mathrm{g}} \bmod T = 0$ **then**
    $P^{\star} \leftarrow$ PGM();
    $(\mathcal{L}_{\mathrm{IAPF}}, \mathcal{L}_{\alpha}) \leftarrow \mathrm{RM}(\mathcal{L}_{\mathrm{APF}}, P^{\star})$;
    $\mathcal{L}_{\mathrm{U}} \leftarrow PI(\mathcal{L}_{\mathrm{APF}}, \mathcal{L}_{\mathrm{IAPF}})$;
    $\mathcal{P} \leftarrow ((\mathcal{P} \setminus \mathcal{L}_{\mathrm{APF}}) \cup \mathcal{L}_{\mathrm{U}})$;
    Update $\mathcal{L}_{\alpha}$;
  end if
  while $|\mathcal{O}| < N$ **do**
    $(c_{\mathrm{p}}, c_{\mathrm{q}}) \leftarrow$ Selection($\mathcal{P}$);
    if $|\mathcal{L}_{\alpha}| > 0$ **then**
      $(c_1, c_2) \leftarrow LDC(c_{\mathrm{p}}, c_{\mathrm{q}}, \mathcal{L}_{\alpha})$;
      $c_1 \leftarrow LDM(c_1, \mathcal{L}_{\alpha})$;
      $c_2 \leftarrow LDM(c_2, \mathcal{L}_{\alpha})$;
    else
      $(c_1, c_2) \leftarrow$ SinglePointCrossover($c_{\mathrm{p}}, c_{\mathrm{q}}$);
      $c_1 \leftarrow$ Mutation(c_1);
      $c_2 \leftarrow$ Mutation(c_2);
    end if
    $\mathcal{O} \leftarrow \mathcal{O} \cup \{c_1, c_2\}$;
  end while
  Call ES($\mathcal{O}$) to evaluate $\mathcal{O}$;
  $\mathcal{P} \leftarrow (\mathcal{P} \cup \mathcal{O})$
  Call FastNon-DominatedSorting($\mathcal{P}$) and Crowding-DistanceAssignment($\mathcal{P}$) to (i) evaluate $\mathcal{P}$, (ii) obtain $\mathcal{L}_{\mathrm{APF}}$, and (iii) select N chromosomes as the new $\mathcal{P}$;
end for

number of objectives, respectively. During each generation, LLM-NSGA-II produces N offspring and subsequently performs an elitist population update by selecting N elite solutions from the merged population of size at most $2N$ solutions.

The primary operations and their corresponding worst-case computational complexities are outlined below.

(1) *Offspring construction*: This process encompasses selection, LDC, and LDM, repeated until the offspring population size $|\mathcal{O}| = N$. Because both LDC and LDM operate at the gene level, assuming a chromosome length of L, a standard implementation that scans or updates loci and executes crossover and mutation requires $O(L)$ time per offspring. Consequently, constructing the entire offspring population requires $O(NL)$. Notably, this complexity aligns with that of conventional crossover and mutation operators in the standard NSGA-II.

(2) *Fast non-dominated sorting:* It is $O\big(J(2N)^2\big)$ on the merged population, where $|\mathcal{P} \cup \mathcal{O}| \leqslant 2N$.

(3) *Crowding-distance assignment:* Applied to at most $2N$ solutions, this process has time complexity $O(J(2N) \cdot \log(2N))$.

(4) *Sorting and selection:* This step applies the crowded-comparison operator ($\prec_n$) to at most $2N$ solutions and runs in $O((2N)\log(2N))$ time.

Therefore, the fast non-dominated sorting procedure dominates the overall per-generation algorithmic overhead of LLM-NSGA-II. The overhead for elitist population update (comprising fast non-dominated sorting on at most $2N$ solutions and crowding-distance assignment) is $O(JN^2)$.

3.5.2 *End-to-end cost*

LLM-NSGA-II systematically invokes ES at each generation to evaluate the offspring set $\mathcal{O}$ of size N. Let $C_{\mathrm{ES}}(k)$ denote the cost of evaluating k candidate solutions with ES, including all J objectives. Then, this cost is defined as $C_{\mathrm{ES}}(N)$. An LLM invocation is triggered every T generations (i.e., when $n_{\mathrm{g}} \bmod T = 0$). Let $\mathbb{I}_t \in \{0, 1\}$ indicate whether generation t is LLM-invoked. In LLM-invoked generations, PI requires evaluating newly generated LLM solutions, with the number of such solutions being at most $|\mathcal{L}_{\mathrm{IAPF}}| = \beta_I$. This imposes an extra evaluation cost of $C_{\mathrm{ES}}(\beta_I)$, assuming these solutions lack prior evaluation. Furthermore, LLM-NSGA-II incurs an additional round of fast non-dominated sorting and crowding-distance assignment in PI costing $O(JN^2)$ in the worst case. Hence,

the per-generation end-to-end cost of LLM-NSGA-II is

$$C_{gen}^{\mathrm{LLM-NSGA-II}} = O(NL) + O(JN^2) + C_{\mathrm{ES}}(N) + \mathbb{I}_t\Big(O(JN^2) + C_{\mathrm{ES}}(\beta_I)\Big). \tag{15}$$

LEAO retains the fundamental end-to-end cost detailed in (15). Besides the LLM-NSGA-II overhead, the LLM-invoked generation incurs LLM inference cost and PGM cost. A single LLM inference call introduces a black-box cost in RM, denoted as $C_{\mathrm{LLM}}(S_{\mathrm{in}}, S_{\mathrm{out}})$, where S_{in} and S_{out} are the numbers of input and output tokens of a prompt in a single LLM call, respectively.

PGM includes (i) prompt assembling, which scales linearly with the prompt size S', and (ii) an LLM-based generation of the reference materials. Thus, the PGM cost is $O(S') + C'_{\mathrm{LLM}}(S'_{\mathrm{in}}, S'_{\mathrm{out}})$, where $C'_{\mathrm{LLM}}(S'_{\mathrm{in}}, S'_{\mathrm{out}})$ is omitted from the per-generation cost if the reference materials are precomputed once and reused across generations.

Therefore, the per-generation end-to-end cost of LEAO is

$$C_{gen}^{\mathrm{LEAO}} = O(NL) + O(JN^2) + C_{\mathrm{ES}}(N) + \mathbb{I}_t\Big(C_{\mathrm{LLM}}(S_{\mathrm{in}}, S_{\mathrm{out}}) + O(S') + O(JN^2) + C_{\mathrm{ES}}(\beta_I)\Big). \tag{16}$$

Amortized over a long run, the average per-generation cost is

$$C_{avg} = O(NL) + O(JN^2) + C_{\mathrm{ES}}(N) + \frac{1}{T}\Big(C_{\mathrm{LLM}}(S_{\mathrm{in}}, S_{\mathrm{out}}) + O(S') + O(JN^2) + C_{\mathrm{ES}}(\beta_I)\Big). \tag{17}$$

When excluding the exogenous costs of ES evaluations and LLM inferences, the intrinsic evolutionary overhead of LEAO remains $O(JN^2)$ per generation, with $O(NL)$ being lower-order under typical NSGA-II settings.

3.5.3 *Comparison with standard NSGA-II*

The per-generation overhead of the standard NSGA-II is determined by the fast non-dominated sorting of $2N$ solutions [40]:

$$O\big(J(2N)^2\big) = O(JN^2). \tag{18}$$

LLM-NSGA-II preserves this asymptotic bound. Although LDC and LDM replace conventional genetic operators, their complexity remains $O(NL)$. Consequently, the elitist population update stage continues to dominate at $O(JN^2)$. In this work, LEAO introduces periodic overhead incurred by the PGM, RM, and PI modules. Specifically, PI requires an additional elitist population update on a subset of at most $2N$ solutions, which increases the constant multiplier of the $O(JN^2)$ term during LLM-invoked generations. Meanwhile, RM introduces an additional inference cost C_{LLM} relative to NSGA-II. In summary, the core evolutionary operations across NSGA-II, LLM-NSGA-II, and LEAO are dominated by $O(JN^2)$ time complexity.

4 Simulation results and discussion

To verify the effectiveness of the proposed LEAO that can better solve the problem **BP**, we conduct verification from three aspects: (1) validation experiments, which evaluate the effectiveness of the LLM-suggestion-driven operators and automatic prompt generation module; (2) comparison with other baseline algorithms; (3) comparison of LLM-only baselines under different $C_{\min}$ values. The numerical calculations were performed using unrounded intermediate values, and the results were rounded for presentation.

4.1 Simulation settings

We adopt an environment based on the indoor building layout shown in Figure 1. Main notations are summarized in Table 2. The settings of the indoor mmWave wireless network and the detailed parameters of LLM-NSGA-II are presented in Table 3. Specifically, we use GPT-5[4)] as our base LLM to create threads and run through its commercial API in the reasoning module and automatic prompt generation module. Every $T = 20$ generations, the reasoning module returns an IAPF together with advantageous alleles. We report cost efficiency as $\varepsilon_{\mathrm{ce}} = 100 \times C/\eta$ (in %), which quantifies coverage probability per unit deployment cost.

4) Model snapshot: GPT-5 (OpenAI, Snapshot 2025-10-04). Accessed: 04-Oct-2025.

Table 2 Main notations and their descriptions.

Parameter	Description
λ, ω	Signal wavelength and wall penetration loss
$T_{\mathrm{r}}/P_{\mathrm{t}}$	RSPT-to-transmit-power ratio
L_{r}	Length of room
W_{c}, W_{d}	Width of corridor and door
α	Signal pathloss exponent
d_{u}^2	Physical size of each RIS unit cell
A	Reflection coefficient of each RIS unit
G	Gain of each RIS unit cell
$\nu \times \hat{\nu}$	Number of unit cells per RIS
$N_{\mathrm{BS}}, \mathcal{P}_{\mathrm{BS}}, \eta_{\mathrm{BS}}$	Number, positions, price of BSs
$N_{\mathrm{RIS}}, \mathcal{P}_{\mathrm{RIS}}, \eta_{\mathrm{RIS}}$	Number, positions, price of RISs
$\mathcal{P}'_{\mathrm{BS}}, \mathcal{P}'_{\mathrm{RIS}}$	Sets BS and RIS positions with $N_{\mathrm{BS}}^{\mathrm{H}}$ and $N_{\mathrm{RIS}}^{\mathrm{H}}$
$\eta_{\max}, \eta_{\min}$	Maximum and minimum network cost
$\mathcal{S}_{\mathrm{BS}}, \mathcal{S}_{\mathrm{RIS}}$	Available coordinate set of BSs and RISs
$\mathcal{I}, \mathcal{J}$	Index set of BSs and RISs
$(x_i, y_i), (\hat{x}_j, \hat{y}_j)$	Coordinate of BSs and RISs
$\varepsilon_{\mathrm{ce}}$	Cost efficiency
n_{w}	Number of penetration wall
d	Transmission distance between BSs and UE
d_1	Transmission distance between RIS and UE
d_2	Transmission distance between BS and RIS
ϕ_m	Fading coefficient between BS and the m-th unit cell
φ_m	Fading coefficient between the m-th unit cell and UE
P_{r}	UE received power
$C_{\min}$	Minimum coverage requirement
C, η	Coverage probability and cost of deployment solution
h	Channel coefficients of direct transmission
g	Channel coefficients of obstructed transmission
$\mathbf{x}, c$	Solution structure and unified structure
N_{g}	Maximum number of generations
n_{g}	Current generation
$\mathcal{P}, N, \mathcal{O}$	Population, population size, and offspring
T	Update period of LLM decisions
$P^\star$	Prompt
$\mathcal{L}_\alpha$	Advantageous alleles generated by LLM
$\mathcal{L}_{\mathrm{APF}}, \mathcal{L}_{\mathrm{IAPF}}$	Chromosomes on APF and IAPF, respectively
$\mathcal{L}_{\mathbf{x}}, \mathcal{L}_c$	Index sets of loci in $\mathbf{x}$ and c, respectively
$\mathcal{L}_{\mathrm{U}}$	Chromosomes on the union of APF and IAPF
e_k, e'_k	Non-mutated and mutated states of the k-th gene
$\alpha_{k,a}$	The a-th advantageous allele for e_k
A_k	Total number of advantageous alleles for e_k
L_k	Locus of e_k
$c_{\mathrm{p}}, c_{\mathrm{q}}$	Two parental chromosomes
c_1, c_2	Two offspring chromosomes
$\mathcal{A}_{\mathrm{p}}, \mathcal{A}_{\mathrm{q}}$	Sets of advantageous-allele loci for c_{p} and c_{q}
ℓ'_{pq}	Crossover point actually chosen
$\mathcal{A}_{\mathrm{pq}}$	Union of $\mathcal{A}_{\mathrm{p}}$ and $\mathcal{A}_{\mathrm{q}}$
$\mathcal{A}'_{\mathrm{pq}}$	Set of loci for maximizing $\mathcal{L}_\alpha$ retention
R_{pq}	Cardinality of set $\mathcal{A}'_{\mathrm{pq}}$
$p_{\mathrm{M}}, p_{\mathrm{C}}$	Mutation and crossover probabilities
$\beta_{\mathrm{M}}, \beta_{\mathrm{C}}, \beta_{\mathrm{P}}, \beta_{\mathrm{I}}$	Hyperparameters for LDM, LDC, and PI

To separately showcase the performance of the LLM-only reasoning method, we construct LLM-only baselines in which the LLM serves as the sole optimizer for generating and refining deployment solutions. Specifically, this

Table 3 Simulation parameters.

Parameter	Value	Parameter	Value	Parameter	Value
λ	0.01 m	ω	-20 dB	$T_{\mathrm{r}}/P_{\mathrm{t}}$	-100 dB
L_{r}	10 m	W_{c}	5 m	W_{d}	2 m
d_{u}	0.005 m	α	2	A	0.9
G	8	ν	50	$\hat{\nu}$	50
η_{BS}	1	η_{RIS}	0.05	ν	50
$\eta_{\max}$	3.5	$\eta_{\min}$	1	$C_{\min}$	0.85
N	30	N_{g}	2000	p_{C}	0.9
p_{M}	0.3	β_{C}	0.5	β_{M}	0.5
β_{I}	3	T	20	–	–

baseline generates solutions in each iteration using only the LLM itself. This setup evaluates solutions directly produced by the LLM. To enable a fair comparison, each LLM-only baseline uses the same LLM API provider and the same frontier model as LEAO. The API settings are also kept identical across all methods, with OpenAI used as the API provider. Specifically, the store parameter is set to true to maintain stateful context from turn to turn, the reasoning effort is set to high, and the maximum number of completion tokens is set to 36 000. The temperature and top-p parameters are left unchanged and use the system-default values. In addition, the LLM-only baseline is prompted with the same prompt directly taken from the automatic prompt generation module of LEAO, rather than relying on a separately designed prompt. Each LLM-only baseline is allocated the same number of LLM API calls as the proposed LLM-NSGA-II approach and is executed for a fixed, consecutive number of iterations that matches the same cost budget owing to API calls. These settings are intended to isolate the capability of the LLM itself when it is used independently of the NSGA-II search process.

We further compare the cost efficiency of the proposed LEAO with a proximal policy optimization (PPO) baseline. PPO represents a DRL approach known for strong optimization performance [41]. Specifically, the PPO policy samples deployment candidates, as LEAO does. Its action vector is organized into three groups. The first two count-control variables determine N_{BS} and N_{RIS}, while the other two groups specify $N_{\mathrm{BS}}^{\mathrm{H}}$ BS candidate index variables and $N_{\mathrm{RIS}}^{\mathrm{H}}$ RIS candidate index variables. Besides, the actor is a multi-layer perceptron (MLP) with a shared trunk and three separate output heads for the count, BS position, and RIS position decisions. This avoids forcing deployment-count decisions and position index decisions to compete in one undifferentiated output layer. Moreover, this PPO baseline uses two critics. The reward critic estimates the positive scalarized reward return. For the cost term, we use a cost critic to predict the normalized network cost return. In particular, the reward first checks whether the coverage probability satisfies $C_{\min}$. Samples below $C_{\min}$ still receive bounded positive rewards, but their rewards remain lower than those assigned to feasible samples. Feasible samples satisfying $C_{\min}$ are then ordered by normalized probability-per-cost efficiency and coverage surplus. The cost critic provides an additional cost-advantage term for the policy update through a coverage-dependent gate, so that cost reduction is emphasized mainly when the sampled solution is close to or above the coverage constraint. Hence, the PPO baseline is not a reward-only implementation that may cause performance degradation. This baseline explicitly separates reward learning and cost learning while retaining the clipped PPO policy update.

For PPO hyperparameters, we set the discount factor to $\gamma = 0.9$. Each rollout buffer contains 256 evaluated deployment solutions from the environment simulator, and the minibatch size is 64. With a learning rate of 1×10^{-2}, this PPO performs five epochs of minibatch updates over each rollout buffer using the clipping parameter $\epsilon = 0.2$. Additionally, the actor trunk and both critics use MLPs with four hidden layers of 256, 128, 128, and 64 units. In these networks, layer normalization and SiLU activations are applied, and Adam is used as the optimizer. We count PPO training progress by iterations to ensure a clear comparison of optimization progress with LEAO. In the implementation, one PPO training iteration denotes one rollout-and-update cycle. At the start of each iteration, the current policy is used to collect a rollout buffer of 256 deployment samples. Each sample is evaluated by the ES as in LEAO. Next, we form the positive reward-return target and the normalized-cost value target, and estimate the corresponding reward and cost advantages using the two critics. For the policy update, the advantage entering the clipped PPO surrogate objective is $\hat{A}_t^{\mathrm{PPO}} = \hat{A}_t^{\mathrm{rew}} - \lambda_{\mathrm{c}} g_t \hat{A}_t^{\mathrm{cost}}$, where $\hat{A}_t^{\mathrm{rew}}$ and $\hat{A}_t^{\mathrm{cost}}$ are the normalized reward and cost advantage estimates, respectively. Moreover, g_t is the coverage-dependent gate and $\lambda_{\mathrm{c}} = 0.35$. We then perform five epochs of minibatch updates on the same buffer, using minibatches of size 64. The actor is optimized with the clipped PPO surrogate objective, the reward critic is trained by regression to the positive reward return, and the cost critic is trained by regression to the normalized-cost return. After these updates, the buffer is discarded, and the next iteration starts from the updated policy. Furthermore, the maximum number of PPO iterations is set to 2000. This gives 256 ES evaluations per rollout-and-update iteration, resulting in a total of $2000 \times 256 = 512000$.

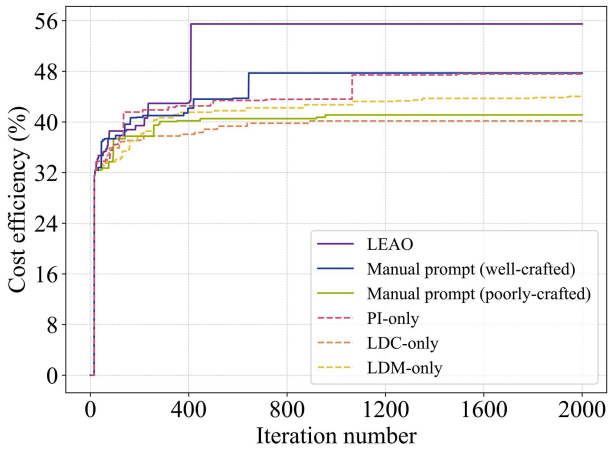

Figure 4 Cost efficiency of the proposed LEAO compared with ablation variants (removing PI, LDC, or LDM in pairs) and manually prompted baselines for the validation of optimization effectiveness.

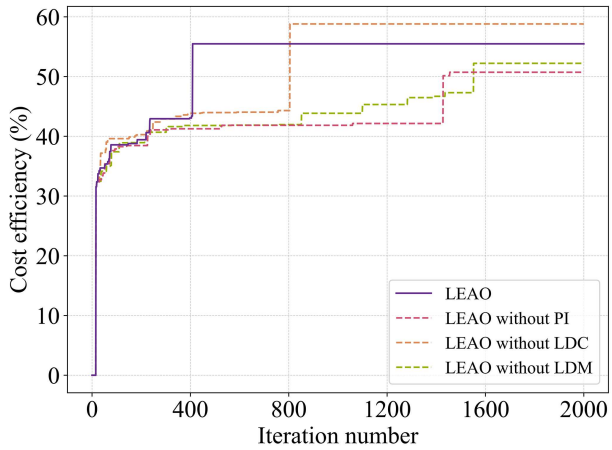

Figure 5 Cost efficiency of the proposed LEAO compared with ablation variants (removing PI, LDC, or LDM singly) for the validation of optimization effectiveness.

For comparison, LEAO performs 30 simulator evaluations per iteration if no additional IAPF re-evaluations are performed. Over the full scheduled run, PPO therefore receives about $512000/60030 \approx 8.5$ times more simulator evaluations than LEAO. This larger per-iteration and total budget makes the comparison conservative with respect to PPO.

4.2 Validation experiments

This section presents validation experiments that evaluate the cost efficiency of the proposed LEAO method across various settings. Figure 4 compares the full LEAO approach with five alternative configurations (three ablation variants and two manually prompted baselines). Additionally, Figure 5 compares the full LEAO approach with the remaining three ablation variants. The ablation variants systematically remove one or more operators of LEAO to reveal their impact on cost efficiency. In particular, we assess configurations that omit single operators (without PI, without LDC, and without LDM) as well as those that omit two operators simultaneously. For comparison, two manually designed prompt configurations are also tested: one uses a well-crafted prompt (referred to as manual prompt (well-crafted)) and the other uses a poorly crafted prompt (referred to as manual prompt (poorly-crafted)). It is observed that among all the considered validation variants, the proposed LEAO is superior to other baselines, where the best optimization performance achieves a cost efficiency of 55.5%. This is because an automatically generated prompt can embody deep domain expertise from the LLM and extract the engineering details from the performance-calculation code. In contrast, the cost efficiency yielded by the LEAO with a well-crafted and manually designed prompt is significantly improved. It shows a relatively small performance gap of about 14.0% compared to the best. Besides, the employment of the automatic prompt generation module significantly enhances optimization in terms of cost efficiency, and the corresponding performance gain is notable (approximately 35.0%) compared to the baseline with a poorly crafted prompt. Moreover, the performance loss between the proposed LEAO and variants that remove one or more LLM-suggestion-driven operators varies. Disabling the LLM-suggestion-driven operators leads to a far larger degradation. The reason is that more LLM-suggested solutions and beneficial BSs/RISs are assembled to produce variations of solutions concentrated on LLM guidance. Overall, the ablation indicates that suggestions from LLMs are critical for guiding the exploration towards promising regions in terms of cost-effective solutions in the search space.

4.3 Comparison of the proposed LEAO with baseline algorithms

To evaluate the effectiveness of the proposed LEAO, four baseline algorithms are simulated.

- NSGA-II: Compared with LLM-NSGA-II in LEAO, NSGA-II is the original version that lacks the LLM-suggestion-driven operators.
- Random deployment (RD): A random deployment approach that only randomly generates solutions to produce cost efficiency.
- Genetic algorithm (GA): GA is a technique that omits the non-dominated sorting method present in NSGA-II.
- PPO: As a state-of-the-art algorithm in DRL, PPO is used as a baseline.

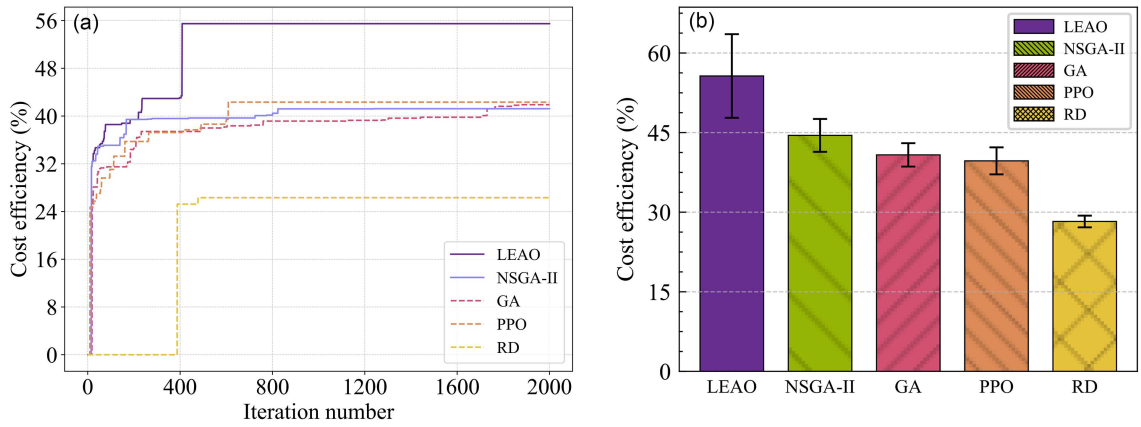

Figure 6 Cost-efficiency comparison between the proposed LEAO using LLM-NSGA-II and the baseline algorithms. (a) Cost efficiency versus iteration number in an independent run. (b) Mean cost efficiency over five independent runs, where each error bar reports average ± standard deviation.

Figure 6(a) shows the achievable cost efficiency versus the iteration number. It is observed that LEAO, NSGA-II, and GA achieve comparable convergence speeds and require comparable numbers of iterations to find the first solution with achievable cost efficiency. This is because these EAs generate new deployment candidates by recombining and mutating selected solutions. Besides, efforts on preserving and constantly improving elites help them move toward solutions satisfying $C_{\min}$. When the proposed LEAO converges, the gain in cost efficiency is about 34.6% and 32.4% compared with the NSGA-II approach and GA method, respectively. This further shows that the intelligent operators can enhance the optimization efficiency by integrating LLM-suggested solutions with evolutionary variation, resulting in superior cost efficiency.

Moreover, the drop in cost efficiency from LEAO to the PPO approach is significant, about 23.7%. Given the ES evaluation budget above, this gap cannot be attributed simply to an insufficient evaluation budget for PPO. Compared to the LEAO budget on ES evaluations, PPO receives a larger allocation both per iteration and in the overall scheduled total. Therefore, the comparison remains conservative for PPO. Instead, the performance gap is mainly due to the different search mechanisms. The PPO approach optimizes a scalarized actor objective. Even with the strengthened reward and cost critics, it learns a search distribution preferred by that scalarization. Unlike the PPO baseline, LEAO uses the LLM-NSGA-II. Its optimization is based on the two objectives, coverage probability and network cost, together with non-dominated sorting and crowding distance. It can therefore maintain multiple candidates considering the trade-off between coverage and cost rather than collapsing the search into one scalar preference. In addition, LEAO preserves elite solutions and refines good feasible regions after they are discovered. Feasible deployments still have to be evaluated by the environment simulator, but once a near-feasible deployment appears, LEAO can retain and refine it in later generations. Lastly, the RD method shows a lower bound on the optimization performance in terms of achievable cost efficiency.

Figure 6(b) reports the cost efficiency obtained from five independent random seeds for each method. Each bar represents the mean value across the five runs, while the error bar denotes the corresponding standard deviation. In line with the high ε_{ce}-performance presented in Figure 6(a), LEAO is robust to stochastic variation, reaching $55.7\% \pm 7.9\%$. By comparison, NSGA-II, GA, PPO, and RD obtain $44.5\% \pm 3.1\%$, $40.8\% \pm 2.2\%$, $39.7\% \pm 2.5\%$, and $28.3\% \pm 1.1\%$, respectively. Notably, the largest discrepancies with respect to LEAO are observed for PPO and RD, with average reductions of 28.7% and 49.3%, respectively. LEAO also outperforms the two EA baselines, NSGA-II and GA, by 25.2% and 36.4%, respectively. Consequently, these results show that the performance gain of LEAO is not only observed in a representative convergence trajectory, but is also preserved on average across independent stochastic runs. The reported means and standard deviations therefore provide additional evidence that the gain is not caused by a single favorable random initialization.

4.4 Comparison with LLM-only baselines under different minimum coverage requirements

Figure 7 shows the achievable cost efficiency under $C_{\min} = 0.65$ and $C_{\min} = 0.85$. To further show the impact of indoor deployment without multi-RISs, we adopt LEAO under the BS-only constraint as two baselines, termed (1) LEAO and $C_{min} = 0.65$ without RISs and (2) LEAO and $C_{min} = 0.85$ without RISs. When the two converge, although the achievable cost efficiency reaches 13.9% and 10.8%, the BS-only baselines not only degrade performance but also increase network cost, reaching 142.9% and 228.6% of the cost budget, respectively. Owing to the use of RIS

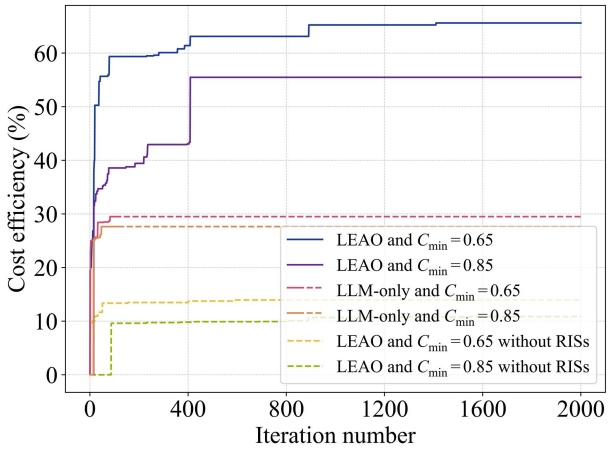

Figure 7 Cost efficiency versus iteration number for the proposed LEAO against LLM-only and BS-only baselines under $C_{\min} = 0.65$ and $C_{\min} = 0.85$.

by the joint deployment, the corresponding performance gains are significant (about 370.5% and 411.4%) compared to the BS-only scheme. To match the same budget of LLM API calls, the portion indicated by dashed lines shows where the LLM-only approach ended when it reached that budget. Compared with the proposed LEAO, it is observed that when the LLM-only approach converges, the loss in cost efficiency performance is approximately 55.0% and 50.2% under $C_{\min} = 0.65$ and $C_{\min} = 0.85$, respectively. The reason is that the LLM, without domain-specific fine-tuning on the task, struggles to achieve remarkable performance in terms of cost efficiency. Furthermore, unlike LEAO under $C_{\min} = 0.65$, when the iteration number approaches 409, the achievable cost efficiency with LEAO under $C_{\min} = 0.85$ converges, consuming approximately 5.1 times as many iterations. This is because $C_{\min} = 0.85$ is more challenging than $C_{\min} = 0.65$. Therefore, Figure 7 validates the effectiveness of the proposed LEAO in the joint indoor deployment task, for which the LLM has not yet improved its ability to generalize.

4.5 Comparison with baselines for avoiding local optima

In this evaluation, we consider three baseline methods in EAs that help escape local optima, and compare them to our proposed method, as shown in Figure 8(a). First, we implement an immigration operator [42] within NSGA-II that periodically injects randomly generated solutions into the population. This approach involves creating new solutions at random, merging them into the existing population, and then applying an additional non-dominated sorting to select the survivors. Second, we evaluate a variant of NSGA-II that omits this extra sorting step and instead forcibly replaces certain individuals with random solutions (termed differential seeds) to escape local optima [15]. Third, we construct a purely random variant of our proposed LEAO by replacing all LLM-suggested solutions with randomly generated ones. To ensure a fair comparison, all three baseline methods perform solution injection at the same update period $T = 20$ used by LEAO. These baselines avoid local optima by introducing new random solutions, allowing comparison with the proposed method in terms of local-optima avoidance. The results show that these three baselines achieve cost-efficiency values comparable to that of the standard NSGA-II, with only minor performance differences. In fact, the proposed LEAO significantly outperforms all three baselines and NSGA-II, indicating a clear advantage of the LLM-based approach. The relatively modest performance of the random baselines can be explained by the limited contribution of random solutions in the complex joint deployment problem of multiple BSs and RISs. In contrast, LEAO leverages the LLM's extensive domain knowledge and reasoning capabilities to suggest high-quality deployment solutions and beneficial BSs/RISs. This guidance yields superior cost-effective solutions and more effective search behavior than the random baselines.

4.6 Performance of GA with the proposed intelligent operators

To further demonstrate the generalizability of the proposed intelligent operators, we integrate them into a standard GA, creating an LLM-enhanced GA (LLM-GA). The comparison baselines in this setting likewise use the same three local-optima escaping strategies described above. In Figure 8(b), the LLM-enhanced GA outperforms all other methods. It achieves an approximately 4.7% improvement in cost efficiency compared to the standard GA. In contrast, the baselines using the other escape mechanisms (immigration, random differential seeds, and random suggestions) achieve a performance similar to that of the standard GA. These results suggest that LLM-enhanced operators provide more targeted and effective search guidance, leading to superior, cost-effective solutions for joint

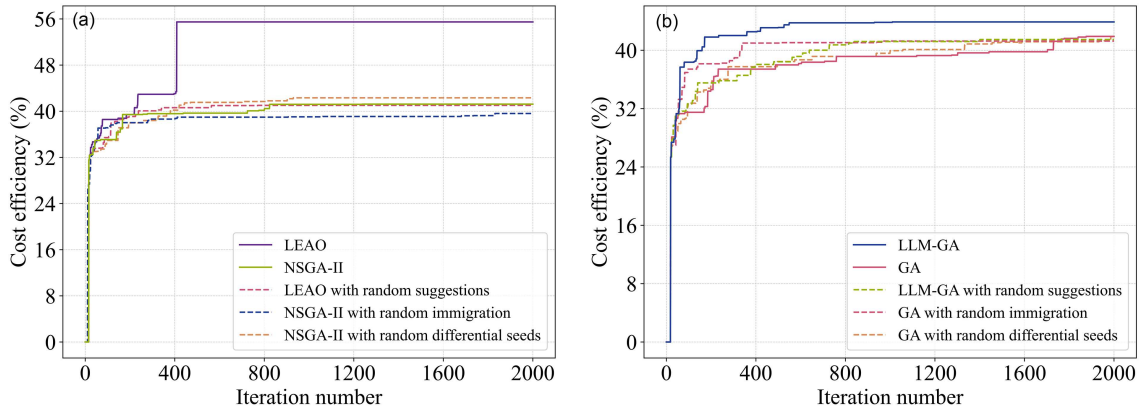

Figure 8 Cost efficiency versus iteration number for (a) LEAO and (b) LLM-GA, each compared with its standard baseline and three local-optima escaping baselines: random immigration, random differential seeds, and random suggestions.

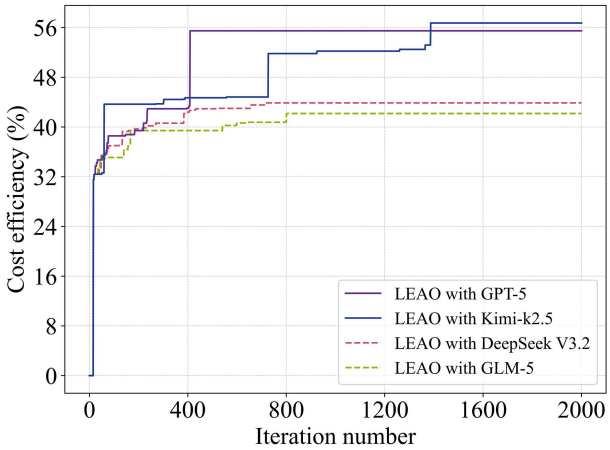

Figure 9 Cost efficiency versus iteration number for LEAO with GPT-5, compared with baseline LEAO variants using DeepSeek V3.2, GLM-5, and Kimi-k2.5 as the LLM.

deployment. In summary, this experiment demonstrates that systematically combining an EA with an LLM can significantly improve optimization efficiency.

4.7 Generalizability across LLMs

We evaluate LEAO across multiple API-accessed LLMs to assess whether its performance gains are robust to the choice of the underlying model. Specifically, we replace the primary LLM used in our main experiments (GPT-5) with three alternative models accessed via their public APIs: DeepSeek-V3.2[5)], GLM-5[6)], and Kimi-K2.5[7)]. Aside from LLM selection, all other parts of LEAO are strictly fixed, including the LLM-NSGA-II, ES, RM, PGM, and stopping criteria. Consequently, the only variable across these experimental conditions is the selected LLM endpoint. All models are used as black-box inference endpoints via their public APIs, without fine-tuning, adapting, or modifying any model weights. For fair comparison, we keep the same prompting template and fix inference parameters across models (temperature, top-p, and maximum token budget).

As illustrated in Figure 9, LEAO implemented with GPT-5 and Kimi-K2.5 achieves comparable cost-efficiency values of 55.5% and 56.7%, respectively, resulting in a gap of 1.3 percentage points (pp). Notably, LEAO with DeepSeek-V3.2 and LEAO with GLM-5 exhibit performance deviations relative to the best-performing configuration (Kimi-K2.5), with observed gaps of approximately 12.8 pp and 14.6 pp, respectively. In summary, the experimental results indicate that the performance enhancements introduced by LEAO persist across state-of-the-art LLMs. These results demonstrate that LEAO can be instantiated using different LLM backends, although the final optimized performance still depends on the models.

5) Model snapshot: DeepSeek-V3.2 (DeepSeek-AI, Snapshot 2026-02-19). Accessed: 19-Feb-2026.
6) Model snapshot: GLM-5 (Zhipu AI (Z.ai), Snapshot 2026-02-19). Accessed: 19-Feb-2026.
7) Model snapshot: Kimi-K2.5 (Moonshot AI, Snapshot 2026-02-19). Accessed: 19-Feb-2026.

5 Conclusion

This paper has studied the joint indoor deployment for an indoor RAN consisting of multiple BSs and RISs. The placement and number configurations for them were jointly optimized to simultaneously minimize the total network cost of BSs and RISs and maximize coverage probability. An LEAO framework was proposed to solve the formulated non-convex and bi-objective optimization problem. This framework benefited from the designed LLM-NSGA-II, which utilized LLM-suggestion-driven operators. These operators contributed to cost efficiency in optimization, as validated in experiments. It also used a prompt generation module developed with in-context learning using the LLM. Simulation results demonstrated the effectiveness of the proposed LEAO framework and its advantages over baseline algorithms. The automatically generated prompt shows superior performance compared to the manually designed prompt, and helps the LEAO efficiently maximize the cost efficiency. The results demonstrate that the approach, which mitigates human intervention, can be a viable alternative to manual prompt design. Furthermore, the effectiveness of the proposed intelligent operators is also validated by adopting them in GA, exhibiting a generalizability and notable performance gain.

Beyond BS and RIS deployment, the LEAO framework represents a significant step toward autonomous, zero-touch optimization in complex engineering domains. The ability to translate natural language constraints into mathematical search guidance allows this architecture to be generalized to other bi-objective problems, such as smart grid management, satellite routing, and industrial logistics. Furthermore, by improving deployment efficiency and reducing the computational "trial-and-error" of manual tuning, our approach contributes to the development of Sustainable AI and energy-efficient 6G infrastructure. Future work will explore the scalability of LEAO in dynamic, real-time environments where network configurations must adapt instantaneously to shifting user demands.

Acknowledgements This work was supported in part by National Key R&D Program of China (Grant No. 2024YFE0200101), UKRI Project RISIR (Grant No. EP/Y023374/1), MSCA IPOSEE Project (Grant No. 101086219), Innovate UK (Grant No. 10099265), and Horizon Europe COVER Project (Grant No. 101086228) (with funding from UKRI, Grant No. EP/Y028031/1). Kezhi WANG acknowledges the support of the Royal Society Industry Fellowship (IF\R2\23200104).

References

1 Ge X, Tu S, Mao G, et al. 5G ultra-dense cellular networks. IEEE Wireless Commun, 2016, 23: 72–79
2 Wu Q, Zhang R. Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts. IEEE Trans Commun, 2020, 68: 1838–1851
3 Hou S, Saqib N U, Chae S H, et al. Optimal placement of reconfigurable intelligent surface for millimeter-wave indoor communication. In: Proceedings of 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), 2022. 698–700
4 Liu X, Liu Y, Chen Y, et al. RIS enhanced massive non-orthogonal multiple access networks: deployment and passive beamforming design. IEEE J Sel Areas Commun, 2021, 39: 1057–1071
5 Fu M, Mei W, Zhang R. Multi-passive/active-IRS enhanced wireless coverage: deployment optimization and cost-performance trade-off. IEEE Trans Wireless Commun, 2024, 23: 9657–9671
6 Encinas-Lago G, Albanese A, Sciancalepore V, et al. A cost-effective RISs deployment to abate the coverage problem in B5G networks. IEEE Trans Wireless Commun, 2024, 23: 15276–15290
7 Zhang J, Blough D M. Optimizing coverage with intelligent surfaces for indoor mmWave networks. In: Proceedings of IEEE Conference on Computer Communications, 2022. 830–839
8 Liu Z, Liu Y, Chu X. Reconfigurable-intelligent-surface-assisted indoor millimeter-wave communications for mobile robots. IEEE Internet Things J, 2024, 11: 1548–1557
9 Huang P Q, Zhou Y, Wang K, et al. Placement optimization for multi-IRS-aided wireless communications: an adaptive differential evolution algorithm. IEEE Wireless Commun Lett, 2022, 11: 942–946
10 Mei W, Zhang R. Joint base station and IRS deployment for enhancing network coverage: a graph-based modeling and optimization approach. IEEE Trans Wireless Commun, 2023, 22: 8200–8213
11 Kong D, Nambala S, Rusek F, et al. Base-station and RIS deployment optimization for indoor coverage enhancement. In: Proceedings of 2023 IEEE Conference on Antenna Measurements and Applications (CAMA), 2023. 246–249
12 Pavai G, Geetha T V. A survey on crossover operators. ACM Comput Surv, 2017, 49: 1–43
13 Zhou H, Hu C, Yuan Y, et al. Large language model (LLM) for telecommunications: a comprehensive survey on principles, key techniques, and opportunities. IEEE Commun Surv Tutorials, 2025, 27: 1955–2005
14 Chang Y, Wang X, Wang J, et al. A survey on evaluation of large language models. ACM Trans Intell Syst Technol, 2024, 15: 1–45
15 Tian H, Han X, Wu G, et al. An LLM-empowered adaptive evolutionary algorithm for multi-component deep learning systems. In: Proceedings of the 39th AAAI Conference on Artificial Intelligence and 37th Conference on Innovative Applications of Artificial Intelligence and 15th Symposium on Educational Advances in Artificial Intelligence, 2025. 39: 20894–20902
16 Wang Y, Afzal M M, Li Z, et al. Large language model as a catalyst: a paradigm shift in base station siting optimization. IEEE Trans Cogn Commun Netw, 2025, 11: 4313–4327
17 Qiu K, Bakirtzis S, Wassell I, et al. Large language model-based wireless network design. IEEE Wireless Commun Lett, 2024, 13: 3340–3344
18 Trigui I, Affes S, Liang B. Unified stochastic geometry modeling and analysis of cellular networks in LOS/NLOS and shadowed fading. IEEE Trans Commun, 2017, 65: 5470–5486
19 Zheng H, Zhang J, Hu H, et al. The analysis of indoor wireless communications by a blockage model in ultra-dense networks. In: Proceedings of 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), 2018. 1–6
20 Müller M K, Taranetz M, Rupp M. Analyzing wireless indoor communications by blockage models. IEEE Access, 2016, 5: 2172–2186
21 Müller M K, Taranetz M, Rupp M. Effects of wall-angle distributions in indoor wireless communication. In: Proceedings of IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications, 2016. 1–5
22 Lee J, Zhang X, Baccelli F. A 3-D spatial model for in-building wireless networks with correlated shadowing. IEEE Trans Wireless Commun, 2016, 15: 7778–7793
23 Niu Y, Li Y, Jin D, et al. A survey of millimeter wave communications (mmWave) for 5G: opportunities and challenges. Wireless Netw, 2015, 21: 2657–2676

24 Tang W, Chen M Z, Chen X, et al. Wireless communications with reconfigurable intelligent surface: path loss modeling and experimental measurement. IEEE Trans Wireless Commun, 2021, 20: 421–439
25 Yildirim I, Uyrus A, Basar E. Modeling and analysis of reconfigurable intelligent surfaces for indoor and outdoor applications in future wireless networks. IEEE Trans Commun, 2021, 69: 1290–1301
26 Li Z, Hu H, Zhang J, et al. Enhancing indoor mmWave wireless coverage: small-cell densification or reconfigurable intelligent surfaces Deployment? IEEE Wireless Commun Lett, 2021, 10: 2547–2551
27 Gao Y, Xiao Y, Lei X, et al. Pricing for reconfigurable intelligent surface aided wireless networks: models and principles. IEEE Netw, 2023, 37: 1–8
28 Bouras C, Kokkinos V, Papazois A. Financing and pricing small cells in next-generation mobile networks. In: Proceedings of Wired/Wireless Internet Communications, 2014. 41–54
29 Huang S, Yang K, Qi S, et al. When large language model meets optimization. Swarm Evolary Computation, 2024, 90: 101663
30 Ma H, Zhang Y, Sun S, et al. A comprehensive survey on NSGA-II for multi-objective optimization and applications. Artif Intell Rev, 2023, 56: 15217–15270
31 Juan A A, Keenan P, Martí R, et al. A review of the role of heuristics in stochastic optimisation: from metaheuristics to learnheuristics. Ann Oper Res, 2023, 320: 831–861
32 Deb K, Goel T. Controlled elitist non-dominated sorting genetic algorithms for better convergence. In: Proceedings of Evolutionary Multi-Criterion Optimization, 2001. 67–81
33 Williams G C. Pleiotropy, natural selection, and the evolution of senescence. Sci Aging Knowledge Environ, 2001, 2001: cp13
34 Long E, Zhang J. Evidence for the role of selection for reproductively advantageous alleles in human aging. Sci Adv, 2023, 9: eadh4990
35 Pang R, Deng Z, Chiu Y. Pareto improvement through a reallocation of carbon emission quotas. Renew Sustain Energy Rev, 2015, 50: 419–430
36 Mao Y, He J, Chen C. From prompts to templates: a systematic prompt template analysis for real-world LLMapps. 2024. ArXiv:2504.02052
37 Manias D M, Chouman A, Shami A. Semantic routing for enhanced performance of LLM-assisted intent-based 5G core network management and orchestration. In: Proceedings of IEEE Global Communications Conference, 2024. 2924–2929
38 Zheng H S, Mishra S, Chen X, et al. Take a step back: evoking reasoning via abstraction in large language models. In: Proceedings of the 12th International Conference on Learning Representations, 2024
39 Crupi G, Tufano R, Velasco A, et al. On the effectiveness of LLM-as-a-judge for code generation and summarization. IIEEE Trans Software Eng, 2025, 51: 2329–2345
40 Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Computat, 2002, 6: 182–197
41 Luong N C, Hoang D T, Gong S, et al. Applications of deep reinforcement learning in communications and networking: a survey. IEEE Commun Surv Tutorials, 2019, 21: 3133–3174
42 Ebrahimi Zade A, Sadegheih A, Lotfi M M. A modified NSGA-II solution for a new multi-objective hub maximal covering problem under uncertain shipments. J Ind Eng Int, 2014, 10: 185–197