Journal of
Marine Science and Engineering

MDPI

Article

Method for Collaborative Layout Optimization of Ship Equipment and Pipe Based on Improved Multi-Agent Reinforcement Learning and Artificial Fish Swarm Algorithm

Hongshuo Zhang , Yanyun Yu *, Zelin Song, Yanzhao Han, Zhiyao Yang and Lang Ti

School of Naval Architecture & Ocean Engineering, Dalian University of Technology, Dalian 116024, China; 22203182@mail.dlut.edu.cn (H.Z.); songzelin@mail.dlut.edu.cn (Z.S.); hanyanzhao@mail.dlut.edu.cn (Y.H.); yzy2022@mail.dlut.edu.cn (Z.Y.); tilang@mail.dlut.edu.cn (L.T.)

* Correspondence: yuyanyun@dlut.edu.cn; Tel.: +86-0411-8470-7485

Abstract: The engine room is the core area of a ship, critical to its operation, safety, and efficiency. Currently, many researchers merely address the ship engine room layout design (SERLD) problem using optimization algorithms and independent layout strategies. However, the engine room environment is complex, involving two significantly different challenges: equipment layout and pipe layout. Traditional methods fail to achieve optimal collaborative layout objectives. To address this research gap, this paper proposes a collaborative layout method that combines improved reinforcement learning and heuristic algorithms. For equipment layout, the engine room space is first discretized into a grid, and a Markov decision process (MDP) framework suitable for equipment layout is proposed, including state space, action space, and reward mechanisms suitable for equipment layout. An improved adaptive guided multi-agent Q-learning (AGMAQL) algorithm is employed to train the layout model in a centralized manner, with enhancements made to the agent's exploration state, exploration action, and learning strategy. For pipe layout, this paper proposes an improved adaptive trajectory artificial fish swarm algorithm (ATAFSA). This algorithm incorporates a hybrid encoding method, adaptive strategy, scouting strategy, and parallel optimization strategy, resulting in enhanced stability, accuracy, and problem adaptability. Subsequently, by comprehensively considering layout objectives and engine room attributes, a collaborative layout method incorporating hierarchical and adaptive weight strategies is proposed. This method optimizes in phases according to the layout objectives and priorities of different stages, achieving multi-level optimal layouts and providing designers with various reference schemes with different focuses. Finally, based on a typical real-world engine room engineering case, various leading algorithms and strategies are tested and compared. The results show that the proposed AGMAQL-ATAFSA (AGMAQL-ATA) exhibits robustness, efficiency, and engineering practicality. Compared to previous research methods and algorithms, the final layout quality improved overall: equipment layout effectiveness increased by over 4.0%, pipe optimization efficiency improved by over 40.4%, and collaborative layout effectiveness enhanced by over 2.2%.

Keywords: collaborative layout optimization; ship engine room layout design; encoding technique; automation design; muti-agent reinforcement learning; artificial fish swarm algorithm

check for updates

Citation: Zhang, H.; Yu, Y.; Song, Z.; Han, Y.; Yang, Z.; Ti, L. Method for Collaborative Layout Optimization of Ship Equipment and Pipe Based on Improved Multi-Agent Reinforcement Learning and Artificial Fish Swarm Algorithm. *J. Mar. Sci. Eng.* **2024**, *12*, 1187. https://doi.org/10.3390/jmse12071187

Academic Editor: Graciliano Nicolás Marichal

Received: 19 June 2024
Revised: 5 July 2024
Accepted: 10 July 2024
Published: 15 July 2024

1. Introduction

With the rapid development of the global shipping industry and the increasing importance of energy efficiency optimization, ship performance optimization, and especially engine room layout design has become crucial for enhancing operational efficiency and safety. The engine room, located at the core of the ship, is a vital part of detailed ship design, directly affecting the overall performance and maintenance convenience of the ship. SERLD encompasses two core aspects: the equipment system and pipe system. The functional attributes and layout priorities of the equipment system are complex and diverse, and its layout is crucial for the ship's stability and propulsion. The pipe system connects these pieces of equipment and transmits essential substances. Collaborative layout problems become highly complex due to the interrelatedness and independence of these two aspects. Traditional methods typically adopt an independent, step-by-step optimization approach. While this simplifies the design process, it usually overlooks the intricate functional connections between equipment and pipe, lacks flexibility, and struggles to adapt to complex and variable practical application scenarios. Currently, research in the SERLD field primarily includes pipe layout research, equipment layout research, and collaborative layout research. The focus of this paper is to utilize reinforcement learning (RL) and optimization algorithms for collaborative research. Unlike traditional optimization algorithms, RL is a novel collaborative optimization technique and is central to this study. Therefore, this section will focus on discussing it in detail. Next, we will provide a specific discussion on the research progress in this field to highlight the contributions of this paper, divided into four main aspects:

1.1. Related Research

1.1.1. Pipe Layout Research

Currently, the collaborative layout of the ship engine room still mainly relies on manual efforts. Existing research mainly unilaterally focuses on either the ship pipe layout problem (SPLP) or the ship equipment layout problem (SELP), including innovations in encoding methods and optimization techniques. In terms of encoding methods, equipment is typically based on a grid-based space, with center-point coordinates indicating position and the maximum square bounding box representing volume [1]. Most scholars focus on pipe layout exploration. For instance, Dong et al. [2] proposed a co-evolution algorithm with fixed grid length encoding to arrange ship pipes. Bian et al. [3] improved the fixed-length grid encoding method by handling grid points during the optimization process. However, grid encoding methods are spatially constrained and require lengthy preprocessing times. To address this, Lin et al. [4] proposed a more flexible high-dimensional vector encoding method. Subsequently, Zhang et al. [5,6] refined the vector encoding method, significantly improving the efficiency of pipe design. However, the continuous data from the vector method sometimes results in unstable decimal deviations. This paper combines the advantages of grid and vector encoding methods to create a stable and efficient encoding method.

In terms of optimization methods, many scholars have conducted in-depth research in the fields of the SPLP and SELP. The SPLP field mainly focuses on innovative applications of optimization algorithms and layout strategies. The earliest applications are deterministic algorithms such as Dijkstra [7], maze algorithms [8], and A* [9]. However, these methods have poor randomness, strong constraints, and long preprocessing times. Subsequently, Dong et al. [10,11], Niu et al. [12], Lin et al. [13], and Zhang et al. [5], respectively, conducted optimization studies on ship pipe layout based on heuristic genetic algorithms (GAs), ant colony optimization (ACO), particle swarm optimization (PSO), and the AFSA. Recently, Ha et al. [14] proposed a pipe layout method that combines optimization algorithms with expert design experience, incorporating expert experience into the evaluation function. Kim et al. [15] introduced a pipe layout method based on reinforcement learning technology, which can dynamically recognize the path environment and find the optimal layout adapting to the current situation. Dong et al. [16] proposed a ship pipe design framework based on NSGA-III, significantly reducing algorithm complexity. In summary, the layout methods in the studies above still have some deficiencies, such as a lack of intelligence, problem adaptability, layout efficiency, and engineering practicality. Extensive surveys and comparisons indicate that heuristic algorithms are very suitable for SPLP research, and the AFSA, as a typical algorithm, has demonstrated strong optimization performance in various path optimization fields [17–19]. Therefore, this paper further explores and refines the current deficiencies in SPLP research based on the AFSA.

1.1.2. Equipment Layout Research

Compared to the SPLP, research on the SELP is relatively scarce, mainly focusing on optimization algorithms and human–machine integration methods. In terms of optimization algorithms, Luo et al. [20] proposed a ship cabin facility layout optimization design method based on improved PSO, improving the efficiency and quality of maintainability layout design. Lee et al. [21] proposed an improved GA, providing a new solution for the multilayer facility layout with internal structural walls and passages. Besbes et al. [22] introduced a new method and mathematical equation based on GA and A*, considering transport paths and effectively reducing the total material handling cost subjected to production-derived constraints. Lee et al. [23] addressed the equipment layout optimization problem for offshore platforms by proposing an efficient layout method that combines a modified iterative deepening search and an improved greedy algorithm, effectively solving practical issues. In terms of human–machine integration, Mallam et al. [24] explored an early design integration method that incorporates human factors and ergonomics knowledge. Meng et al. [25] proposed a human reliability analysis (HRA) method more suitable for cabin equipment layout optimization and developed a cabin equipment layout optimization solution platform based on the GA. Zhang et al. [26] proposed a human cognitive reliability–cognitive reliability and error analysis method, combined with A* and a GA, to solve ship cabin equipment layout optimization problems considering human factors, minimizing human error probability.

1.1.3. Collaborative Layout Research

It is worth noting that the independent layout studies described in Sections 1.1.1 and 1.1.2 only achieve local optimization for SERLD. Their independent linear summation does not represent the overall optimal collaborative result, primarily due to the following two shortcomings: (1) Lack of consideration for overall layout factors, with equipment layout having higher priority, which significantly affects the collaborative research on subsequent pipe layout. (2) In practical design, this independent SERLD method requires engineers to spend considerable time coordinating, and the results are not optimal. Therefore, collaborative research on SERLD has gradually emerged in recent years. Jiang et al. [27] were the first to study the collaborative layout of equipment and pipes, proposing an evaluation function based on PSO and ACO that considers only equipment relevance, pipe length, and the number of bends. Haris et al. [28] researched the collaborative layout of equipment and pipes for LNG tanks based on computational fluid dynamics (CFD), considering more collaborative factors. Furthermore, Gunawan et al. [29] conducted research on the collaborative layout problem of engine rooms based on the GA, adding considerations for the cost and length of pipe systems in bulk carriers of different sizes. Recently, Zhang et al. [30] proposed a collaborative layout method based on reinforcement learning and the A* algorithm, making SERLD more intelligent. In summary, current SERLD research mostly remains at the two-dimensional level, lacking consideration of overall environmental factors and engineering practicality. It is worth noting that most existing studies consider equipment functional attributes and priorities, combining expert experience to set layout zones and sequences. This paper will continue to follow this approach.

To address the current issues, we recently proposed a novel SERLD method based on heuristic algorithms [6], filling the gap in SERLD research and achieving collaborative layout optimization across multiple levels in three-dimensional space, significantly enhancing engineering practicality. However, our subsequent research found that while heuristic algorithms are relatively stable for the SPLP, as layout scenarios become more complex and the number of variables increases, they struggle to handle the SELP accurately and stably, in terms of parallel problems or orientation problems, etc. Ultimately, this leads to difficulties in convergence or falling into local optima. In practical engineering, SERLD needs to consider more layout requirements and collaborative actions. Additionally, previous research required substantial time to identify collaborative layout zones.

1.1.4. Novel Intelligent Collaborative Technology

As indicated in Section 1.1.3, current collaborative methods or strategies are mostly based on traditional optimization algorithms or engineering software. These methods have numerous limitations and cannot meet the detailed layout requirements and handle large variables. In recent years, artificial intelligence technologies, particularly deep learning (DL) and RL, have developed rapidly. Increasingly, scholars are applying these technologies to solve complex optimization problems, offering valuable solutions for various engineering applications. Due to RL's ability to train models quickly and adapt to changing environments, it is particularly suitable for applications involving multiple variables and continuous feedback loops, such as path planning or SELP-type combinatorial optimization problems [31]. Moreover, researchers have verified that RL is more suitable than optimization algorithms for combinatorial collaborative optimization problems [32], and multi-agent reinforcement learning (MARL) can optimize more objectives, making it more adaptable to SERLD issues. Currently, many researchers apply different MARL algorithms in complex collaborative optimization problems, such as MAQ-learning (MAQL) [33], Wolf-PHC [30], Qmix [34], and MADDPG [35], among others. In this study, the actions and state spaces of the equipment are discrete, making the MAQL algorithm highly suitable for such problems. Due to its ease of handling, efficiency, and stability, MAQL is widely applied in many fields. Adeogun et al. [36] established an MAQL method based on limited sensing information, transforming resource selection in 6G in-X Subnetworks into a multi-objective optimization problem, effectively achieving dynamic resource allocation. Zhou et al. [37] used the MAQL algorithm to address the recharging scheduling problem of electric automated guided vehicles in container terminals, minimizing operational delays by generating reasonable recharging plans. Wang et al. [38] addressed the adaptive traffic signal control problem in large-scale scenarios using an improved MAQL algorithm, contributing to the transportation field. In summary, MAQL offers strong flexibility and adaptability, allowing for the targeted training of layout models with various functionalities according to specific needs. This paper will continue to explore applications based on MAQL to address the collaborative optimization problems in SERLD.

In conclusion, this study, aiming at the issues revealed by current optimization algorithms and layout strategies in SERLD research, explores a hybrid algorithm that combines reinforcement learning and heuristic algorithms for collaborative layout. In response to traditional independent layout strategies, this study further proposes a comprehensive collaborative evaluation function and multiple collaborative layout strategies based on the novel hybrid algorithm. This aims to achieve a more efficient, intelligent, and robust collaborative layout method, solving more complex practical engineering problems.

1.2. Innovative Contributions and Engineering Significance

This study addresses the deficiencies in optimization algorithms and layout strategies in previous SERLD collaborative research by integrating artificial intelligence technology and heuristic algorithms for the first time. The aim is to explore an efficient and practical collaborative algorithm and layout strategy based on the different layout objectives and engineering constraints of equipment and pipe systems, significantly enhancing the automation and layout efficiency of SERLD. The feasibility and engineering value of the proposed method is ultimately verified through practical engineering cases. In summary, the main innovations of this paper are as follows: (1) Pipe layout aspect—A precise and stable adaptive trajectory-based vector encoding method with hybrid concepts is proposed. Based on this method, an improved AFSA (ATAFSA) with multiple optimization strategies is further introduced. (2) Equipment layout aspect—An adaptive guided multi-agent reinforcement learning (AGMAQL) algorithm is proposed, along with an MDP framework suitable for equipment layout problems, including state space, action space, and collaborative reward mechanisms, and so on. (3) Collaborative layout aspect—A layout method with a hierarchical layout strategy and an adaptive collaborative weight strategy is proposed, based on the collaborative algorithm (AGMAQL-ATA). The collaborative evaluation function is also refined.

The rest of this paper is organized as follows: Section 2 introduces the SERLD concept, constraints, objectives, and parameterization methods. Section 3 introduces the collaborative layout method based on the improved AGMAQL-ATA. Section 4 validates and discusses the effectiveness of the proposed layout algorithm and method through practical engine room cases. Finally, Section 5 contains the conclusion of this study.

2. SERLP Formulation

2.1. Parametric Processing and Layout Concept

To effectively address the SERLD problem and ensure computational efficiency, the layout must be conducted based on parametric modeling. Figure 1 outlines an overview of the entire SERLD process, including environmental space parametric modeling, equipment layout, and pipe layout. In the figure, E in Module B represents the abbreviation for equipment, while S and E in Module C represent the start and end points of the pipes, respectively. Module A in Figure 1 indicates that in the initial stage, the most fundamental and necessary step is to parametrize the layout problem, with the maximum boundaries in space being the deck and bulkheads. For the equipment layout problem, the primary goal is to obtain the optimal scheme within the feasible layout zone by considering factors such as safety, relevance, and engineering requirements. Each equipment entity is modeled using the typical bounding box method, representing its volume with the minimum and maximum diagonal coordinate points $\{(X_{min}, Y_{min}, Z_{min}), (X_{max}, Y_{max}, Z_{max})\}$, a method that greatly facilitates algorithmic computation [13]. The position of each piece of equipment is indicated by its central coordinates (e.g., E3 in Module B). In this paper, each piece of equipment is treated as an agent, and the optimal overall layout is achieved by continuously translating and rotating within the feasible layout zone based on the AFSA and MAQL. Pipe layout is based on equipment positions and impacts the ship's operational efficiency, safety, and convenience, among other factors. The pipe model, as shown in Module C, connects related pieces of equipment and consists of start points and intermediate bend nodes. The goal of pipe layout is to find the optimal path within the restricted space and constraints, primarily including single-pipe and branch pipe problems. Additionally, the space includes both prohibited and guiding energy zones that influence the layout of equipment and pipes. Prohibitive energy zones include obstacles or hazardous zones while guiding energy zones include the surrounding space of bulkheads, equipment, already laid pipes, and specially designated zones, among others.

For the SERLD collaborative problem, as shown in Module D of the figure, there are two levels of layout issues, such as the parallel layout and mixing pipe layout problems noted in the figure. During the collaborative layout process, changes in equipment positions affect pipe paths, and these paths, in turn, impact the efficiency of equipment operation. Therefore, it is necessary to effectively link and jointly optimize both aspects under engineering specifications and layout requirements to achieve a truly collaborative layout. Furthermore, this study simplifies the collaborative problem hierarchically, designing reasonable layout strategies and evaluation methods for each level from multiple perspectives. Ultimately, by adjusting weight parameters, multiple optimal collaborative layout alternatives can be provided to designers.

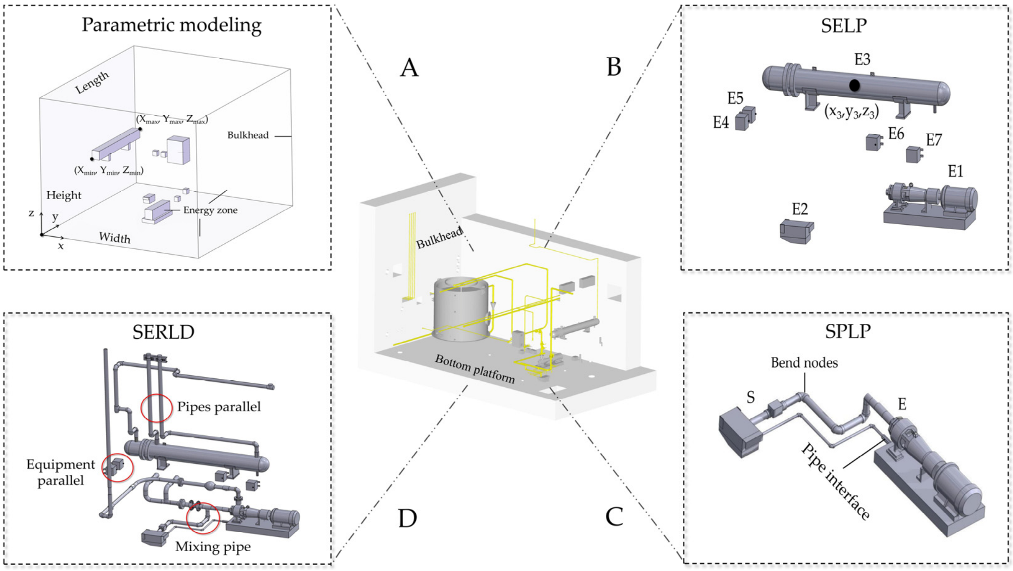


Figure 1. Parametric analysis and layout principles for the SERLD problem: Module A: initial modeling diagram; Module B: SELP diagram; Module C: SPLP diagram; Module D: SERLD diagram.

2.2. Hybrid Encoding Expression

The reasonable encoding of equipment and pipe is fundamental to intelligent layout. This study considers a suitable reinforcement learning environment by discretizing the equipment layout zone into a grid, where the equipment explores by moving along the grid. The grid division precision is based on the side length of the smallest volume of equipment. Figure 2 is a standard two-dimensional example of a grid-based engine room layout. The initial layout range must adhere to basic constraints [39], avoiding excessive randomness. The figure illustrates various constraint requirements, including location restrictions, distance limitations, and special layout requirements. Equipment in different layout zones is distinguished by different colors in the figure. Figure 2b illustrates a poorly designed layout case, where red markers indicate violations of layout requirements, including interferences, failure to meet distance limitations, and non-compliance with parallel layout specifications. As described in Section 1.1, the current literature on layout methods takes into account the actual engine room environment and divides the initial layout range for equipment. This paper first considers basic division principles such as space utilization, engine room stability, and functional relevance. Based on key positions like the central axis and passageways, the initial feasible layout zones are defined [6]. As shown in Figure 2, the space is divided into two large feasible layout zones, followed by the allocation of specific collaborative layout zones for each piece of equipment. As noted in Section 1.1, expert experience is indispensable. To better align with practical engineering layout standards, enhance human–machine interaction, and reduce problem complexity, this study predefines the collaborative layout zone for each piece of equipment during the initial layout phase, based on layout requirements and expert experience. Generally, these zones constitute about 40% of the feasible layout zone to which they belong.

Pipes differ from equipment, as the interface positions are located on the equipment surface, with the layout goal primarily focusing on finding the optimal path nodes between two interfaces within a fixed space. Initially, researchers used a simple grid approach for space discretization, but it was challenging to determine an appropriate resolution, and the space shape constrained it. In recent years, efficient vector encoding methods have emerged, based on continuous space processing. These methods optimize by controlling the trend of vector bend nodes, offering strong flexibility and efficiency. The path expression based on this encoding method is shown in Equation (1). However, relying solely on algorithmic optimization of random vector points is inefficient for pipe routing. Figure 3 illustrates the principles of the vector encoding method, where square shapes represent equipment, and points represent vector nodes. S and E denote the start and end points. Unnecessary space disturbances can affect vector point trends, as shown in Figure 3a, where black vector points outside the feasible zone hinder optimization, while the white points are the ideal effective points. Recently, we proposed a Manhattan trajectory-based vector encoding method (red points in Figure 3b), which maximizes the use of effective space [6]. However, this trajectory can be obstructed by obstacles (such as the red circles in the figure), which can impede the movement of vector points. Additionally, current vector encoding methods still require manually specifying the number of bend nodes needed, lacking intelligence.

$$P = [S, v, E], v \in R \tag{1}$$

where P represents the pipe path, S and E represent the start and end points of the path, and v represents the vector nodes, i.e., bend nodes of the path. The positions of these vector nodes cannot exceed the layout zone R.

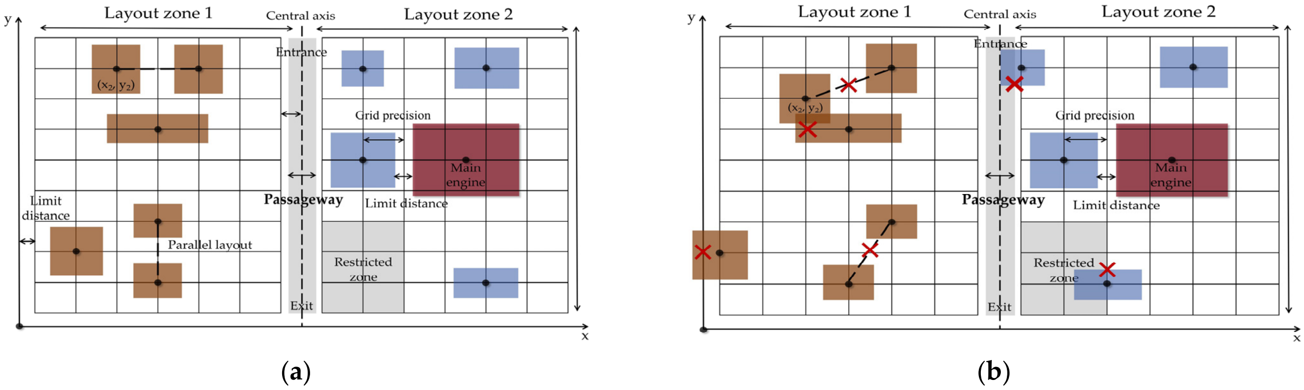


Figure 2. A two-dimensional example of a grid-based engine room layout: (**a**) constraint illustration; (**b**) poor layout.

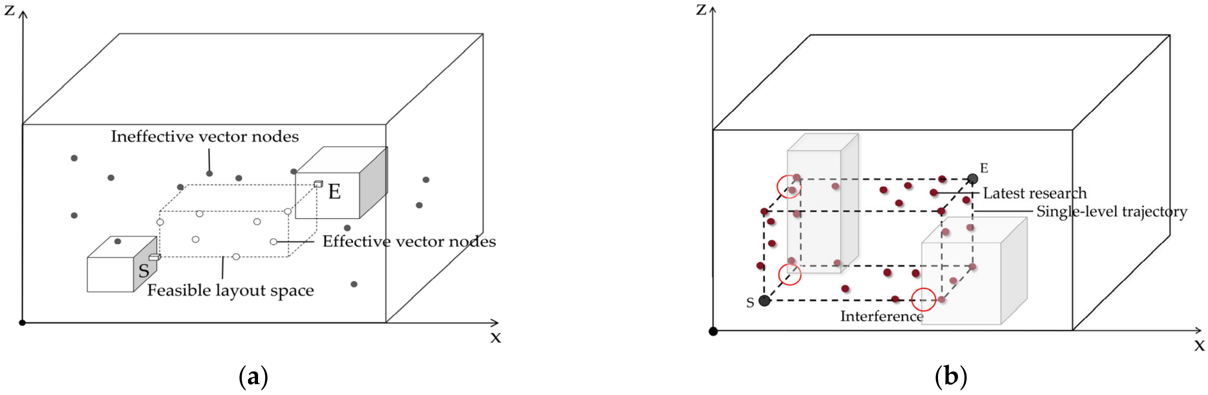


Figure 3. Schematic diagram of the vector encoding method: (**a**) distribution of different vector nodes; (**b**) recent research.

Addressing the shortcomings of current pipe encoding methods, this study combines the advantages of grid and vector methods to propose an adaptive trajectory vector encoding method. This method focuses on the fundamental constraints of pipe layout by adaptively finding the optimal Manhattan trajectory to guide the vector points' trend. The initialized vector points avoid interference from obstacles and non-orthogonal guidance, and this encoding method can autonomously determine the number of bend nodes, overcoming the fundamental drawbacks of previous encoding methods. The specific principle is shown in Figure 4. Figure 4a demonstrates the generation method of Manhattan trajectories in different dimensional spaces. During the optimization process, the trajectory adaptively adjusts according to the environment. As shown in Figure 4b, the first trajectory optimization is first performed between the start point S and the end point E, generating an initial trajectory (grey trajectory). When encountering obstacle interference (red circles in the figure), a new start point, S′, is determined to generate the next orange trajectory (from S′ to E), while retaining the previous SS′ trajectory. Further interference results in generating a blue trajectory (from S″ to E). This process is repeated until overall optimal adaptive trajectories are found that are free from obstacle interference, known as the elite trajectory. This example involved three optimizations using both 3D and 2D trajectory optimization methods. Finally, as seen in Figure 4c, after removing the ineffective trajectories affected by interference, three better trajectory types are obtained through integration. According to the hierarchical division principles described in Section 3.1.1, the trajectories in this example are divided into three levels. After evaluation and ranking, the grey trajectory with the fewest bend nodes is deemed the best and set as level 1, while the yellow and blue trajectories are set as level 2 and level 3, respectively. After identifying the optimal trajectory for each level, initial vector points are allocated around the bend nodes along these trajectories. The distribution space of the vector points is centered around the nodes, occupying about 10% of the engine room space. The initial vector points must not be located within obstacles or exceed the maximum range. Different levels are allocated different numbers of vector nodes, and to ensure flexibility, some vector points are reserved for random optimization.

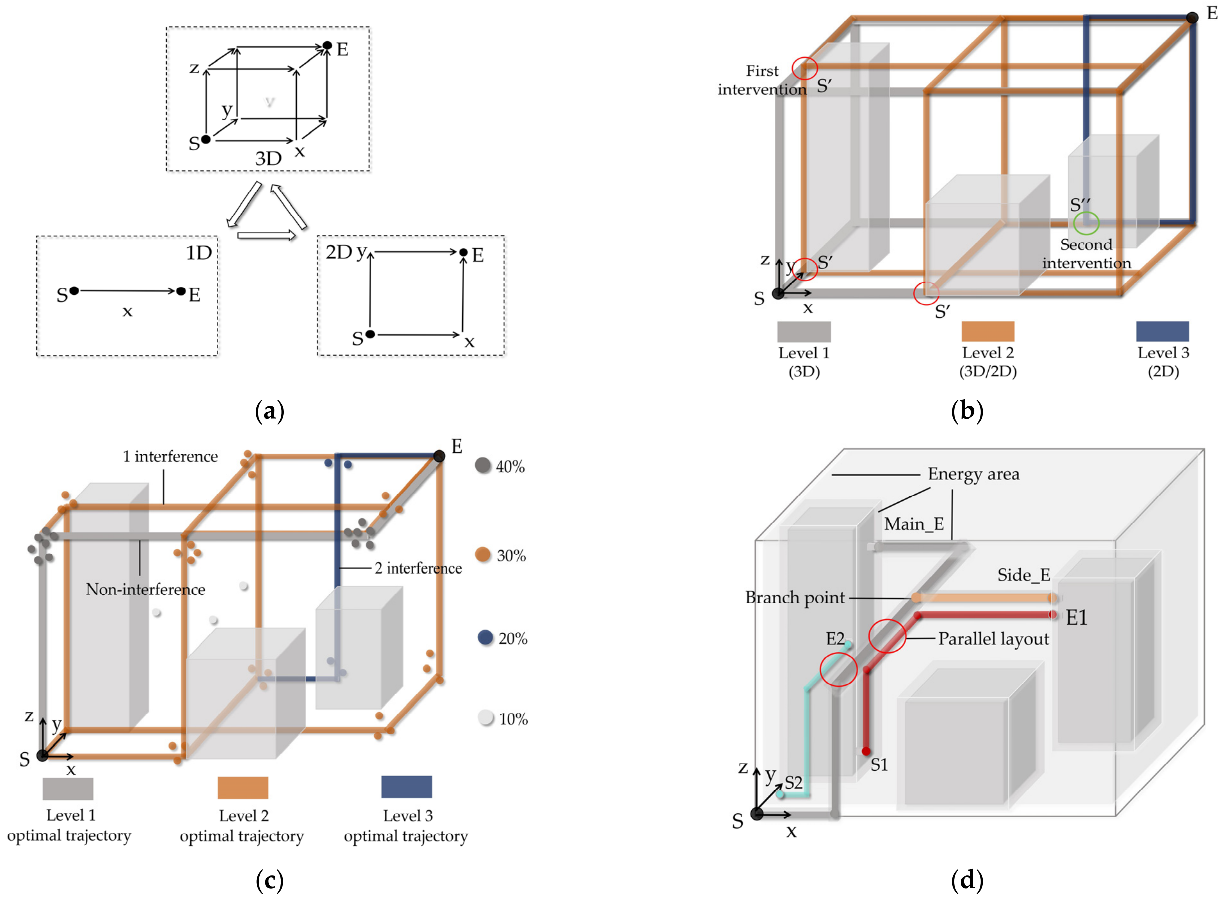


Figure 4. Principle of the adaptive trajectory vector encoding method: (**a**) Manhattan trajectories in three dimensions; (**b**) optimization level display; (**c**) optimal trajectory and distribution of vector points; (**d**) example of mixing pipe layout.

Additionally, the pipe layout problem also includes scenarios involving branch pipe and single-pipe mixed layouts. As shown in Figure 4d, a branch pipe consists of a gray main pipe and yellow side pipes, where Main_E and Side_E represent the endpoints of the main and side pipes, respectively. S1, E1, S2, and E2 denote the start and end points of other single pipes. The layout process for the main and side pipes follows the same procedure as for the layout of the single pipes, utilizing the adaptive trajectory encoding method. It is important to note that when laying out branch pipes, the position of the main pipe and the arrangement order of the side pipes must be determined based on the interface distance and pipe diameter [40]. First, the main pipe is laid out, and then the previously proven effective orthogonal projection method is used to sequentially determine the starting points of each side pipe from the main path [5]. When addressing multi-pipe layout problems, it is necessary to predetermine the layout priorities. The laid-out pipes are considered obstacle zones and are surrounded by a new energy zone. Pipes with lower priority, in addition to considering the energy zones around the bulkheads and equipment, should ideally be laid out parallel to the energy zones of higher-priority pipes to meet engineering layout requirements [11] (red circles in the figure). The figure shows an optimal mixing pipe layout scheme.

2.3. Optimization Objectives and Constraints

The objectives and constraints of the SERLD problem consist of two parts: pipes and equipment. Achieving the optimal overall design result requires addressing objectives and requirements at two levels. Before algorithm exploration, evaluation parameters need to be initialized. A reasonable engine room layout evaluation function is key to achieving objectives. Currently, there are established evaluation methods in the SERLD field [4,6,25], as shown in Equation (2). To ensure the standardization of this study, adaptive adjustments are made based on existing mature evaluation functions. These adjustments consider the stability, functional relevance, safety, engineering practicality, convenience, and economy of the engine room layout to achieve the collaborative goals of SERLD. The specific objectives and constraints at each level are as follows:

$$\begin{gathered}\min F(w\_l) = l_1 \times \min F(e\_l) + l_2 \times \min F(p\_l) \\ l_1 + l_2 = 1\end{gathered} \tag{2}$$

where $F(w\_l)$ represents the evaluation function for the overall collaborative layout, aiming for a minimum value. It consists of the pipe evaluation function $F(p\_l)$ (detailed in Section 2.3.1) and the equipment evaluation function $F(e\_l)$ (detailed in Section 2.3.2), weighted accordingly. p_l denotes all pipe paths and e_l denotes the overall equipment layout scheme. l_1 and l_2 are normalized weight coefficients.

2.3.1. Pipe Layout Evaluation

The optimization objectives for pipe layout are as follows:

Objective 1: Minimize the path length. This objective is achieved by calculating each segment of the adjacent path, with the calculation equation as $f_{le} = \sum_{i=1}^{N} L(v_i, v_{i+1})$.

Objective 2: Minimize the number of bends. This objective is calculated using three adjacent bend nodes, with the calculation equation as $f_{be} = \sum_{i=1}^{N} B(v_{i-1}, v_i, v_{i+1})$.

Objective 3: Maximize the traversal through guiding energy zones. In complex mixing pipe layouts, the energy zones can be flexibly adjusted based on engineering requirements. This objective determines the upper limit of the layout. It is achieved by calculating the positional relationship between the path and the energy zones, with the calculation equation as $f_{en} = \sum_{i=1}^{N} P(v_i, v_{i+1}, e\_list)$. The variable e_list represents the list of energy zones.

The constraints for pipe layout are as follows:

Constraint 1: Pipe paths must maintain orthogonal direction. This constraint is achieved by calculating the orthogonality between two nodes, with the calculation equation as $f_{or} = \sum_{i=1}^{N} O(v_i, v_{i+1})$.

Constraint 2: Pipe paths must not cross obstacles or restricted zones. This constraint is achieved by calculating the positional relationship between the path and the restricted zones, with the calculation equation as $f_{in} = \sum_{i=1}^{N} S(v_i, v_{i+1}, r\_list)$. The variable r_list represents the list of restricted zones.

Constraint 3: The mixing pipe must be laid out in the specified order. Adjacent pipes or adjacent bend nodes must adhere to the minimum distance requirement.

Based on the above objectives and constraints, the evaluation function for pipe layout can be derived as follows: The function $F(p\_l)$ and its weight proportions are consistent with the literature. Here, p_1 to p_5 represent the weights of each evaluation sub-item, specifically 10:1:0.01:0.1:0.001.

$$\min F(p\_l) = p_1 \times \min f_{in}(p\_l) + p_2 \times \min f_{or}(p\_l) + p_3 \times \min f_{le}(p\_l) + p_4 \times \min f_{be}(p\_l) - p_5 \times \max f_{en}(p\_l) \quad (3)$$

2.3.2. Equipment Layout Evaluation

The optimization objectives for equipment layout are as follows:

Objective 1: Ensure minimum imbalance in torque difference. The overall equipment layout significantly impacts the safety of the engine room. The torques on either side of the central axis need to maintain a certain balance. The calculation equation for this objective is shown below, where G_i represents the weight of equipment i, y_i represents the longitudinal coordinate of equipment i, and W represents the width of the engine room.

$$f_{ba}(e\_l) = \sum_{i=1}^{N} G_i(y_i - W/2) \quad (4)$$

Objective 2: Ensure functional relevance and safety. Equipment with close functional relationships should be positioned at short distances to improve operational efficiency and space utilization. Simultaneously, considering safety and engineering practicality, equipment should be positioned near exits to allow for personnel to quickly evacuate in case of an emergency. The calculation equation for this objective is shown below, where D_{ij} represents the relevant distance between equipment. If pipe connections exist, D_{ij} denotes the distance between the interfaces; otherwise, it denotes the distance between the equipment's center points. τ_{ij} represents the relevance coefficient, ranging from 0 to 1. S_i represents the distance from equipment i to the exit, and φ represents the safety coefficient, ranging from 0.1 to 0.5. The setting principles for τ and φ can be found in Reference [6].

$$f_{re}(e\_l) = \sum_{i=1}^{N} \sum_{j=1}^{N} D_{ij} \times \tau_{ij} + \varphi \times S_i \quad (5)$$

Objective 3: Ensure ease of pipe layout. To achieve a truly collaborative layout, the layout of equipment should consider the locations of pipe interfaces. Fewer obstacles between equipment connected by pipes is preferable, and the pipe interface points should be as close or parallel as possible. Equation (5) also reflects this objective.

The constraints for equipment layout are as follows, with detailed calculations available in Reference [30]:

Constraint 1: Equipment must not interfere or overlap with each other (Equation (6)). The square of the overlapping volume is used as a penalty value.

$$P_1(e\_l) \rightarrow \{E_i \cap E_j | \forall i, j \in N, i \neq j\} \quad (6)$$

Constraint 2: Equipment must maintain a specified minimum distance from each other and bulkheads (Equation (7)). The square of the violated volume is used as a penalty value.

$$P_2(e\_l) \rightarrow \{\{|E_{ic} - E_{jc}| > D\} \cap \{|E_{ic} - H_c| > D\} | \forall i, j \in N, i \neq j, c = (x, y, z)\} \quad (7)$$

Constraint 3: Equipment with a similar volume or functional nature should be arranged in parallel (Equation (8)). If the positions are not parallel, the square of the minimum deviation length is used as a penalty value. Similarly, if the orientations are not parallel, the square of the deviation volume is used as a penalty value, excluding the z-axis direction.

$$P_3(e\_l) \rightarrow \{E_{ix} = E_{jx} \cup E_{iy} = E_{jy} \cup E_{iz} = E_{jz} | V_i = V_j, \forall i, j \in N, i \neq j\} \quad (8)$$

Based on the above objectives and constraints, the evaluation function for equipment layout is derived as follows:

$$\min F(e\_l) = s_1 \times \min f_{ba}(e\_l) + s_2 \times \min f_{re}(e\_l) + s_3 \times \min f_{co}(e\_l) \tag{9}$$

where the approach for $F(e\_l)$ is consistent with that in Reference [6], aiming for minimization while maintaining an order of magnitude balance with $F(p\_l)$. The main difference in this study is the adjustment of the pipe calculation part, eliminating unnecessary calculations for pipes in the equipment layout layer and optimizing the relevance function $f_{re}(e\_l)$, making the problem simpler and more precise. Ultimately, based on the previously proposed weight coefficient normalization principles, s_1 is set to 0.02, s_2 to 0.78, and s_3 to 0.2. $f_{co}(e\_l)$ represents the constraint term for equipment layout, integrating Equations (6)–(8).

3. Collaborative Layout Method Based on Improved AGMAQL-ATA

This section focuses on the core content of this study, including the logic of the improved algorithm and the collaborative layout design method for the engine room. The algorithm aspect mainly involves the coordinated improvement of MAQL and the AFSA, which incorporates multiple layout strategies. The pipe layout is primarily centered on the AFSA, implemented through an adaptive trajectory-based encoding method, adaptive strategy, and parallel strategy. The equipment layout is primarily centered on MAQL, implemented through grid encoding methods, hierarchical concepts, and expert knowledge, and employs guided, adaptive, and greedy strategies. Subsequently, considering layout attributes, optimization objectives, and specification requirements, an overall collaborative layout procedure based on AGMAQL-ATA and collaborative optimization concepts is proposed. The specific content is as follows:

3.1. Ship Pipe Layout Based on Improved ATAFSA

3.1.1. Principles of ATAFSA

The AFSA, a heuristic bionic algorithm, was proposed in 2003, mimicking the collective behavior of fish to achieve optimization [41]. As shown in Figure 5, the AFSA has four core behaviors: preying (selecting a better individual within the search vision to move towards), swarming (comparing with the center evaluation value of nearby individuals to decide whether to move), following (comparing with the best evaluation value of nearby individuals to decide whether to move), and random behavior (randomly moving a step when no better direction is found within the search vision). Each individual moves based on the search vision (S_V) and search step length (S_S), and the optimization process is controlled and guided by the congestion factor (C_F) and the number of search attempts (*Try_num*). S_S must be less than S_V. These core parameters are fundamental to the algorithm's calculations. Our previous research has demonstrated the effectiveness of AFSA in pipe layout problems [5]. To overcome the issues of the AFSA's tendency to get trapped in local optima and its performance instability, this study continues to adopt a parameter-adaptive strategy (as shown in Equation (10)) and a scouting optimization strategy (as shown in Equations (11) and (12)).

$$\begin{cases} \tau(t) = \tau_{\min} + \dfrac{1-\tau_{\min}}{1+randn\times\left(\frac{(t-t_0)^{\alpha}}{T}\right)^{\beta}} \\ S\_V(t+1)/S\_S(t+1) = S\_V(t)/S\_S(t) \times \tau(t) \end{cases} \tag{10}$$

where t_0, t, and T, respectively, denote the initial, current, and maximum iteration numbers, and α and β are coefficients controlling the descent rate. *randn* denotes a random number used for fine-tuning the curve, while $\tau_{\min}$ denotes the specified minimum lower limit. The values of α, β, and *randn* in this study are consistent with those in the literature. The core parameters S_V and S_S are adaptively adjusted during iterations using these equations.

$$\begin{cases} X_{i/next} = X_i + Randn \times (b_{\max} - b_{\min}) \\ X_{\mathrm{i}/next} = X_i + Randn \times S\_S \times \frac{X_a - X_i}{\|X_a - X_i\|} \end{cases} \tag{11}$$

$$X_{\mathrm{i}/next} = \begin{cases} N(X_i, S) \\ S = m \times (b_{\max} - b_{\min}) \end{cases} \tag{12}$$

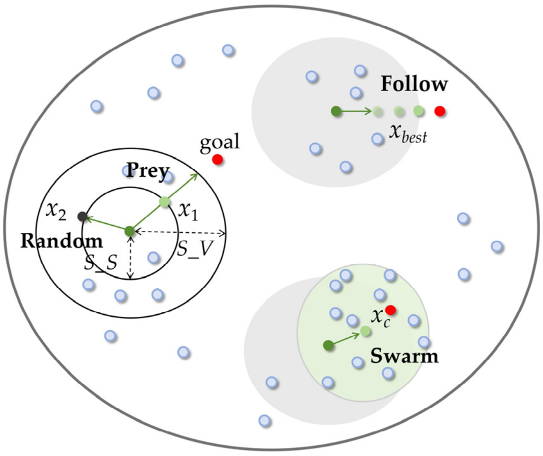


Figure 5. Schematic diagram of AFSA optimization.

Equation (11) represents the random mutation scouting operation. When trapped in a local optimum, several individuals are selected around the optimal individual, and each individual X_i randomly performs the following two operations: (1) a large random update of the position within the layout zone; (2) random movement towards any individual X_a in the space. Equation (12) represents the elimination mutation scouting operation, which involves performing Gaussian mutations on all saved local optimal individuals and replacing the same number of inferior ones. The selected individual X_i undergoes Gaussian mutation based on the standard deviation S. $Randn$ represents a random coefficient, ranging from 0 to approximately 700 in this study. $Randn \times S\_S$ is the random step length, controlling the amplitude in X_i changes. Consistent with previous research, the initial search vision range is approximately 20% of the feasible zone, and the maximum step length variation amplitude is about 10% [5]. $b_{\max}$ and $b_{\min}$ represent the maximum and minimum dimensions of the space, and m represents the mutation control coefficient. This strategy is executed when no better path is found after four consecutive iterations.

Additionally, to accommodate the adaptive trajectory encoding method and enhance optimization accuracy and stability, this study proposes a parallel optimization strategy. Considering computational complexity and optimization stability, a maximum of four layout levels are set. In the initial stage, all explored trajectory types are evaluated and ranked from best to worst, selecting the three optimal trajectory types to correspond to the optimal three layout levels. To balance layout flexibility, a random layout level of 4 is set, which does not rely on any trajectory for layout but randomly generates vector point positions, with the number of bend nodes also randomly generated based on the top three levels. As shown in Figure 6, there are four hierarchical division situations in practical layout problems. The allocation principle for the initial vector population number adjusts accordingly in different situations. Section 2.2 presents a standard situation: when the number of trajectory types is three or more (Situation 4), the first to fourth levels are allocated 40%, 30%, 20%, and 10% of the total population, respectively. When there are fewer than three trajectory types (Situations 1 and 2), the number of random vectors increases correspondingly.

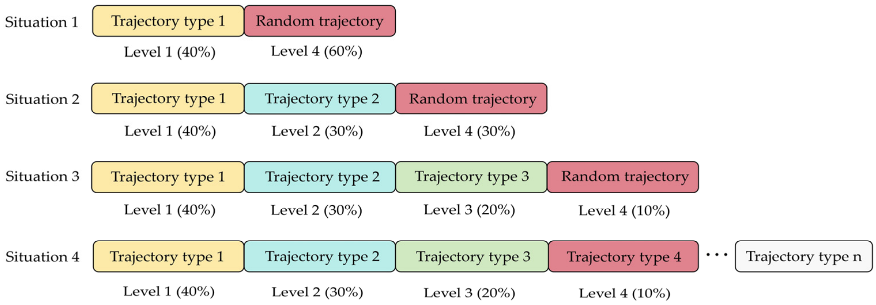


Figure 6. Principle of hierarchical division.

It is important to note that different trajectory types have different numbers of bend nodes, making different path types incompatible during calculations. To avoid calculation mismatches, the initial population is divided into four levels corresponding to the four layout levels. Subsequently, all possible better paths are further explored based on the optimal trajectories of different types. Figure 7 illustrates the parallel optimization process, showing that the algorithm's optimization operations are executed in parallel at the four levels. During the global calculation iteration process, each level operates independently. Ultimately, by evaluating and screening the optimal paths for each level, the optimal pipe layout is output.

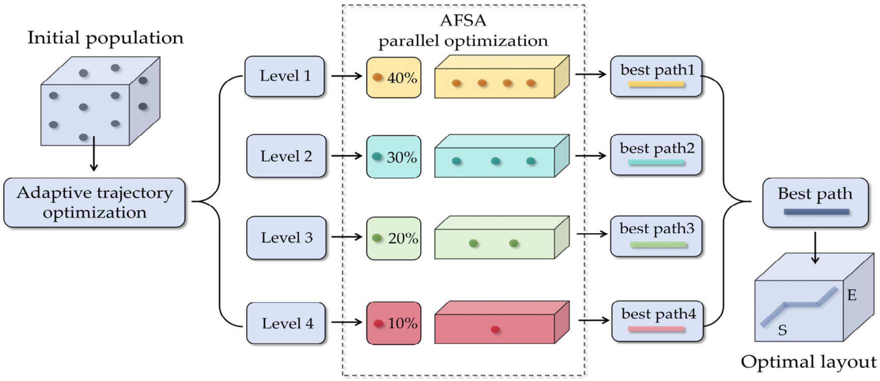


Figure 7. Parallel optimization process.

3.1.2. Pipe Layout Process

Based on the above improvement strategies, the pipe layout process of the ATAFSA is shown in Figure 8. Note that when laying out multiple pipes, the order of layout can be manually set or automatically determined by the system based on the equipment level or pipe diameter. When arranging branch pipes, it is necessary to first find the optimal main pipe path.

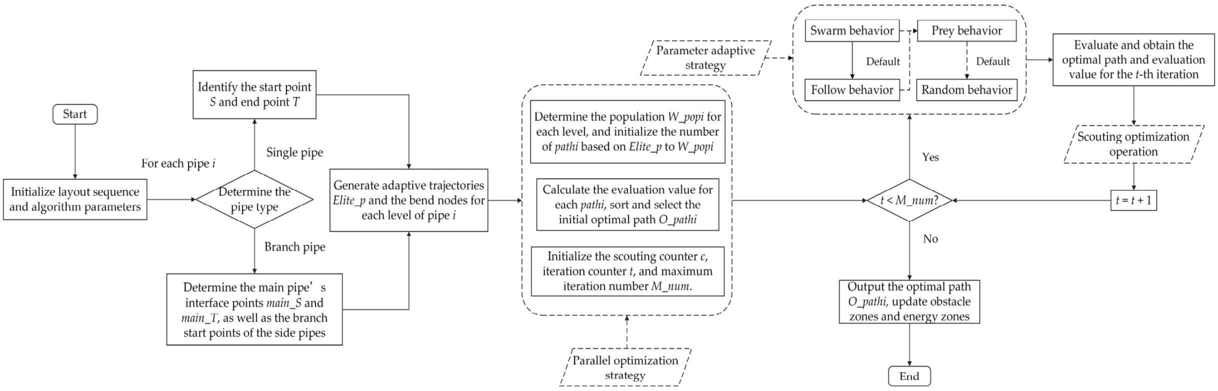

Start
Initialize layout sequence and algorithm parameters
For each pipe i
Determine the pipe type
Single pipe
Identify the start point S and end point T
Branch pipe
Determine the main pipe’ s interface points main_S and main_T, as well as the branch start points of the side pipes
Generate adaptive trajectories Elite_p and the bend nodes for each level of pipe i
Determine the population W_popi for each level, and initialize the number of pathi based on Elite_p to W_popi
Calculate the evaluation value for each pathi, sort and select the initial optimal path O_pathi
Initialize the scouting counter c, iteration counter t, and maximum iteration number M_num.
Parallel optimization strategy
Parameter adaptive strategy
Swarm behavior
Prey behavior
Default
Default
Follow behavior
Random behavior
Evaluate and obtain the optimal path and evaluation value for the t-th iteration
Scouting optimization operation
t = t + 1
t < M_num?
Yes
No
Output the optimal path O_pathi, update obstacle zones and energy zones
End

Figure 8. ATAFSA layout flowchart.

3.2. Ship Equipment Layout Based on Improved AGMAQL

3.2.1. Basic Principles of MAQL

The essence of reinforcement learning lies in learning automatic decision-making through continuous trial-and-error training in the environment, thereby aligning decisions with expectations and moving towards maximum reward feedback. Its main components include agent, state, action, reward, and environment [42]. Q-learning (QL), proposed by Watkins, is a model-free, value-function-based, off-policy reinforcement learning algorithm [43]. Compared to other reinforcement learning algorithms, QL has fewer parameters, a simple structure, and an easy training method, making it widely used. This method primarily relies on state sequences in the MDP to self-adjust and select the optimal actions [44]. As shown in Figure 9a, during the learning process, the agent interacts with the environment to obtain the current state signal s_t and selects an action a_t based on this signal. The state of the environment changes accordingly. The agent evaluates the reward r_t received in this state and decides whether to transition to a new state. QL continuously updates through rewards and punishments to ultimately achieve the optimal $Q(s, a)$ function, i.e., the Q-table (as shown in Table 1). By updating the Q-table, the expected total discounted reward is maximized, resulting in the optimal action selection strategy. The update iteration equation for QL is based on the concept of temporal difference (TD), as shown below.

$$\begin{cases} Q^{t+1}(s_t, a_t) \leftarrow Q^t(s_t, a_t) + \alpha\left(r(s_t, a_t, s_{t+1}) + \gamma \max\limits_{a_{t+1}\in A} Q^t(s_{t+1}, a_{t+1}) - Q^t(s_t, a_t)\right) \\ \alpha \in (0,1) \\ \gamma \in (0,1) \end{cases} \tag{13}$$

where $Q^t(s_t, a_t)$ represents the Q-value for taking action $\mathrm{a_t}$ in state s_t. $r(s_t, a_t, s_{t+1})$ denotes the immediate reward when the environment transitions from state s_t to state s_{t+1} after action a_t. A represents the entire action space. α is the learning rate. γ is the discount factor, indicating the degree of future influence. To balance QL's exploration and exploitation abilities, α is set to 0.05 and γ is set to 0.9.

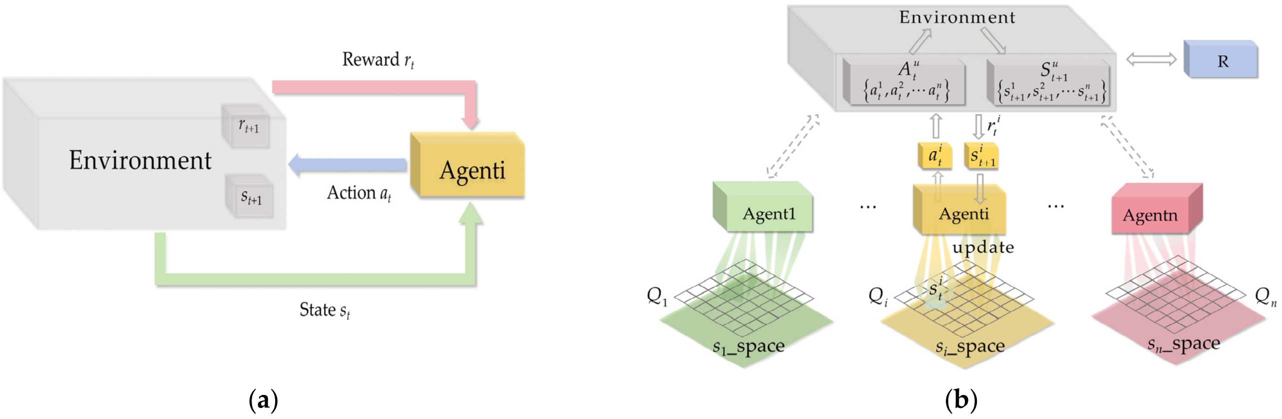


Figure 9. The schematic diagram of MAQL: (**a**) basic QL model; (**b**) MAQL model.

Table 1. The Q-table of QL.

State	**Action**			
	a_1	a_2	$\cdots$	a_n
s_1	$Q(s_1, a_1)$	$Q(s_1, a_2)$	$\cdots$	$Q(s_1, a_n)$
s_2	$Q(s_2, a_1)$	$Q(s_2, a_2)$	$\cdots$	$Q(s_2, a_n)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
s_m	$Q(s_m, a_1)$	$Q(s_m, a_2)$	$\cdots$	$Q(s_m, a_n)$

For the collaborative layout problem in this study, a single-agent algorithm is ineffective. Instead, a multi-agent algorithm aims to combine multiple single agents to optimize a common goal, addressing complex collaborative problems [45]. This approach is characterized by cooperation, parallelism, and distribution, making it particularly suitable for collaborative layout issues. MAQL, one of the most classic MARL algorithms, is widely applied to discrete problems. In this study, MAQL employs a classical centralized training method, with its model shown in Figure 9. At discrete time t, the states and actions of all agents jointly form a combined state space $S^u = s^1 \times s^2 \times \ldots \times s^i$ and a combined action space $A^u = a^1 \times a^2 \times \ldots \times a^i$. Each agent's Q-table is shown in Table 2. Compared to QL, MAQL enables communication and cooperation among agents. Each agent not only understands its state but also accesses the overall states of all agents, allowing for an evaluation based on the entire environment to select the optimal action.

Table 2. The Q-table of MAQL.

State	**Action**		
	a_1	a_2	a_3
$S_1^u \leftarrow \{s_1^1, s_1^2, \cdots s_1^n\}$	$Q(S_1^{\mathrm{u}}, a_1)$	$Q(S_1^{\mathrm{u}}, a_2)$	$Q(S_1^{\mathrm{u}}, a_3)$
$S_2^u \leftarrow \{s_2^1, s_1^2, \cdots s_1^n\}$	$Q(S_2^{\mathrm{u}}, a_1)$	$Q(S_2^{\mathrm{u}}, a_2)$	$Q(S_2^{\mathrm{u}}, a_3)$
$\vdots\ \vdots\ \ldots\vdots$	$\vdots$	$\vdots$	$\ddots$
$S_{m^n}^u \leftarrow \{s_m^1, s_m^2, \cdots s_{\mathrm{m}}^n\}$	$Q(S_{m^n}^{\mathrm{u}}, a_1)$	$Q(S_{m^n}^{\mathrm{u}}, a_2)$	$Q(S_{m^n}^{\mathrm{u}}, a_3)$

3.2.2. Design of the MDP Framework

1. Design of the state space

The state space is defined as the set of information received by each agent i from the environment at time step t. Representing the agent state using grid coordinates is a classical method [30,46]. After dividing the layout zone into grids, each agent can move along the grid. Equation (14) and Figure 10 illustrate the design principle of the state space. The state space for each agent consists of discrete central coordinates (x_{ic}, y_{ic}, z_{ic}) and orientation O_i. Each agent has its own feasible layout zone $w_i \times l_i \times h_i$. Additionally, based on the orthogonality requirements of the pipe path, each agent has four vertical orientations along the bottom platform: 0°, 90°, 180°, and 270°. The orientation of the interface points affects the start or end positions of the pipes. When the direction changes, the values of a, b, and c will be exchanged, and their signs may change.

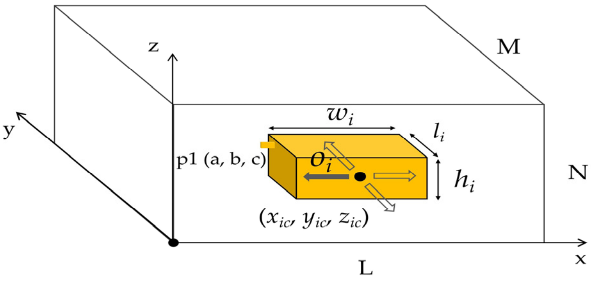


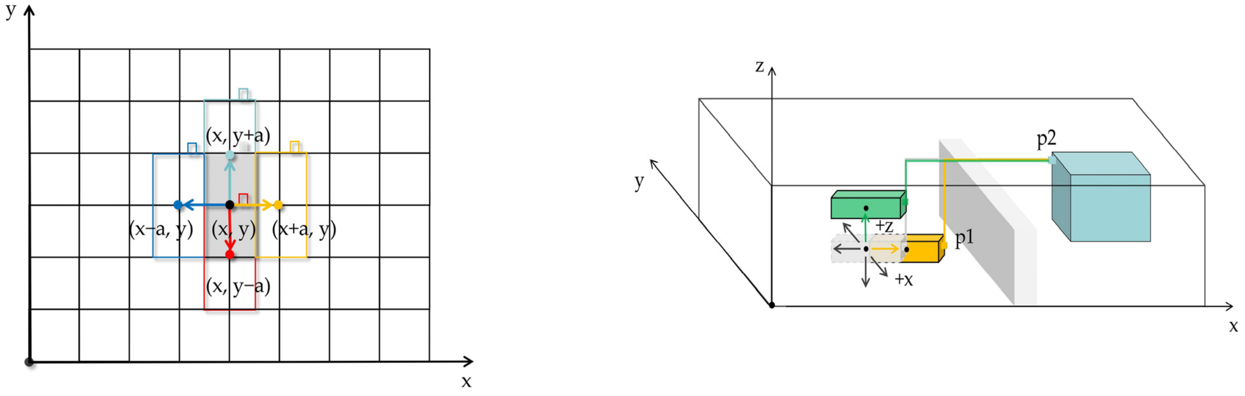
Figure 10. Schematic diagram of state space.

$$\begin{gathered} S_{\mathrm{t}} = \{s^1, s^2 \cdots s^n\}, n = 1,2,3 \cdots i \\ s_t^i \to \begin{bmatrix} C_t^i \\ O_t^i \end{bmatrix} = \begin{bmatrix} (X_c, Y_c, Z_c) \\ Orientation \end{bmatrix}, \begin{cases} (X_c, Y_c, Z_c) \in W \\ Orientation \in (0°, 90°, 180°, 270°) \end{cases} \end{gathered} \tag{14}$$

2. Design of the action space

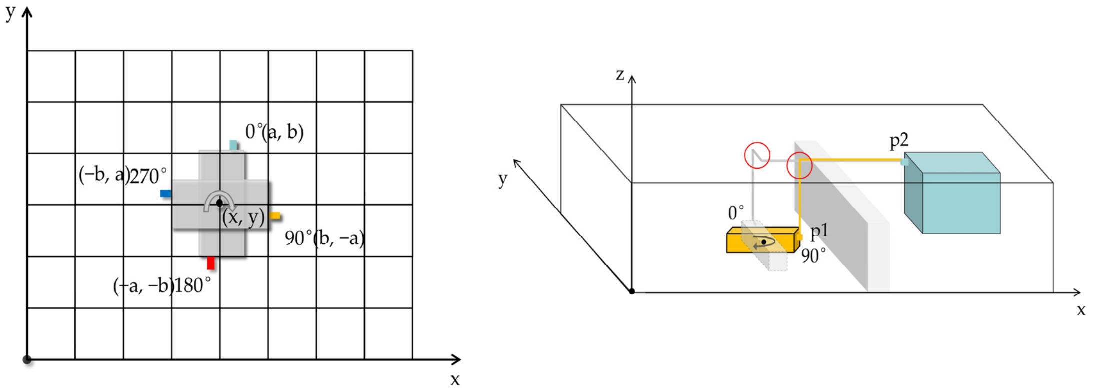
$$A_t^i \rightarrow \begin{bmatrix} M_t^i \\ R_t^i \\ C_t^i \end{bmatrix} = \begin{bmatrix} (-dx, +dx, -dy, +dy, -dz, +dz) \\ (0^\circ, 90^\circ, 180^\circ, 270^\circ) \\ x_c, y_c, z_c, o_c \end{bmatrix} \tag{15}$$

The action space in this study is also discretized, divided into translation, rotation, and constraint actions, as shown in Equation (15). Translation actions involve moving the equipment position along the grid points of the x, y, and z axes to find better layout schemes. Figure 11 shows four translation actions in the xy-plane. When an agent selects a translation direction, it needs to predict all possible movement directions in that state and filter the positions after translation using Equation (2) to determine the optimal direction. It should be noted that there are multiple fixed structural platforms in the engine room, and the z-coordinate of some equipment (such as equipment located on the bottom platform of the engine room) cannot be changed [47]. Rotation actions involve changing the direction of pipe interfaces to facilitate pipe layout, as shown in Figure 12. In practical engineering, equipment usually rotates parallel to the xy-plane to avoid being upside down. When an agent selects a rotation direction, it also needs to perform evaluation predictions to determine the optimal direction. As shown in Figure 12b, after evaluation and filtering, the agent selects the 90° orientation, significantly reducing bends and the length of the pipe.



(**a**) (**b**)

Figure 11. Schematic diagram of translation action: (**a**) 2D space; (**b**) 3D space.



(**a**) (**b**)

Figure 12. Schematic diagram of rotation action: (**a**) 2D space; (**b**) 3D space.

Constraint actions are primarily designed to flexibly meet the hard constraints of equipment layout while considering engineering practicality. These actions include translation and rotation, primarily used to address situations where equipment is stuck in a locally optimal state that does not meet basic constraints for an extended period and cannot transition. According to engineering specifications, certain equipment must meet special layout requirements such as parallelism or symmetry and must strictly adhere to minimum distance constraints. Relying solely on optimization algorithms and basic actions cannot reliably handle certain special local optimal situations and layout requirements. In each state, the agent must first determine whether it is in a special situation. If so, it selects a constraint action to minimally adjust the equipment's state to meet layout constraints and escape the local optimum while ensuring minimal impact on the overall optimization goal. Figure 13 illustrates four typical situations requiring constraint adjustments. Figure 13a,b illustrates situations where the equipment fails to meet the minimum distance requirements. Figure 13c,d illustrates situations where associated agents of the same volume require parallel layout. It can be observed that through constraint actions, equipment E1 is minimally adjusted to meet the basic constraints and achieve the desired layout effectiveness.

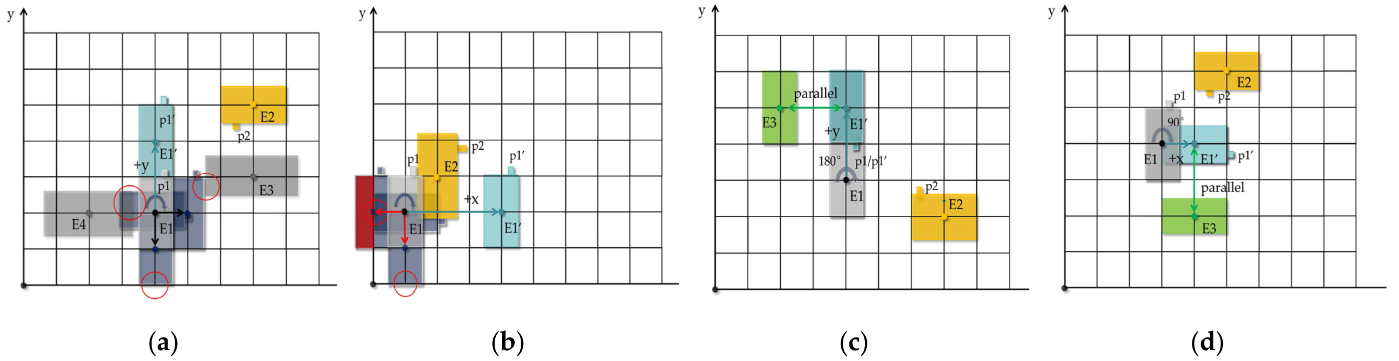


Figure 13. Schematic diagram of constraint action: (**a**) distance constraint 1; (**b**) distance constraint 2; (**c**) parallel constraint 1; (**d**) parallel constraint 2.

3. Design of the reward mechanism

In addition to designing the action and state spaces, MAQL requires a rational reward mechanism to evaluate states, guiding the agents toward better state transitions. For the multi-agent reinforcement learning layout problem, it is necessary to balance the consideration of each equipment's state to minimizing the evaluation function value. This paper uses Equation (2) as the basis for setting up the reward mechanism, as shown in Equation (16). This equation includes a penalty term, *l_space*, primarily to prevent agents from exceeding the feasible layout zone, with a penalty value set at 100,000. If the evaluation value $F_t(w\_l)$ of the joint state at each time step is smaller than the historical optimal evaluation value $F_{\min}(w\_l)$, the reward r is 1, prompting a state transition and updating $F_{\min}(w\_l)$. Otherwise, the reward r is 0, and the state remains unchanged.

$$r \rightarrow \begin{cases} \min F_t(\mathrm{w\_}l) = \min F_t(w\_l) + l\_space \\ \quad 1, e^{(F_{\min}(w\_l) - F_t(w\_l))} > 1 \\ \quad 0, e^{(F_{\min}(w\_l) - F_t(w\_l))} \leq 1 \end{cases} \tag{16}$$

3.2.3. Principles of AGMAQL

1. Guided reduction in state space

For MAQL, the design of the MDP framework is crucial, as it directly determines the solvability of the problem. The MDP must be established within finite state and action spaces, ensuring that the value function estimate converges to the optimum with probability 1. As shown in Table 2, the collaborative problem in this paper considers a vast number of joint states for each Q-table, and the action search range for the agents is also quite limited. When MAQL cannot find a better position within the reachable search zone, the state does not transition. This presents significant challenges for MAQL's training process, making it difficult to traverse all states and actions, ultimately resulting in only locally optimal solutions. Although traditional MAQL can randomly initialize states within the space, the limited search space and overly blind initialization positions cause MAQL to repeatedly fall into local optima. Consequently, the cumulative reward curve fails to converge, preventing the achievement of a consistently rising ideal result.

To address this issue, MAQL needs to be provided with effective assistance to guide the agents in focusing on the most valuable directions for continuous training. In the initial training phase, this study uses the AFSA, known for its strong global optimization capabilities, as an aid to narrow the search space, thereby avoiding the exploration of ineffective zones. Figure 14 simulates the learning and optimization process of an agent within the layout zone in a two-dimensional space, where higher brightness indicates better evaluation values. It can be seen that the entire communication optimization process is illustrated by the yellow circles in Figure 14. Initially, due to the AFSA's strong optimization capability, MAQL quickly identifies the optimal zone. Later, as the AFSA's fine optimization capability is limited by its step size parameters (indicated by the blue circles in Figure 14), the local optimization is primarily reliant on MAQL. It is important to note that during the communication learning process, data processing is required to ensure that each operation is based on grid points. For example, in Figure 15, the blue scheme represents the joint position obtained by AFSA before correction, while the red state is the joint position after grid processing based on the nearest distance principle.

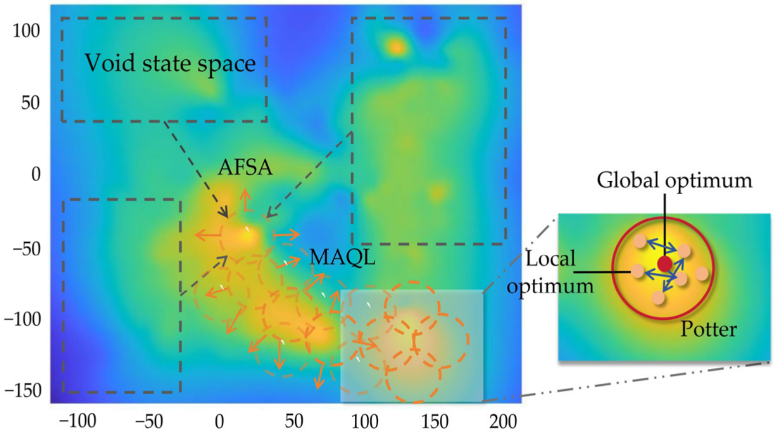


Figure 14. AGMAQL optimization process.

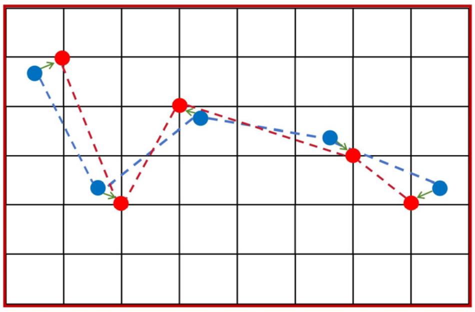


Figure 15. Schematic diagram of grid data processing.

2. Adaptive changes in action space

As discussed in Section 3.2.2, MAQL struggles to escape local optima due to its limited exploration space. Although guided reduction in state space ensures stable MAQL convergence, it does not guarantee efficient attainment of the global optimal solution every time. Figure 16 shows the state space of an agent in the later stages of optimization. The gray point represents the agent's state, and the red point represents the global optimal target. Traditional single-step actions can only explore the green area, and since the states in the green area receive a reward of 0, the state does not transition. Therefore, the searches conducted in this case are ineffective. To avoid this situation, this study designs a variable-scale step length strategy. When the number of ineffective explorations reaches a certain threshold, the step length adaptively increases according to Equation (17), thus continuously expanding the search area (as shown in the figure, the area enclosed by the blue points has already covered the red target point), exploring more states, and increasing the probability of finding the optimal solution.

$$step_{new} = step_{ini} + \min(\lfloor (num\_t - \theta)/\delta \rfloor, step_m), num\_t > \theta \times m\_st \tag{17}$$

where $step_{ini}$ represents the standard step length, m_st represents the maximum number of learning steps per episode, and θ is the limiting coefficient, approximately 0.1 times m_st. num_t indicates the number of ineffective explorations. When num_t exceeds $\theta \times m\_st$, the variable-scale step length strategy is executed. δ is the scaling factor, approximately 0.05 times m_st, which controls the rate of step length change. $step_m$ is the maximum step length, set to three times $step_{ini}$. The values of θ and δ are based on grid precision and can be adaptively adjusted to ensure algorithm performance.

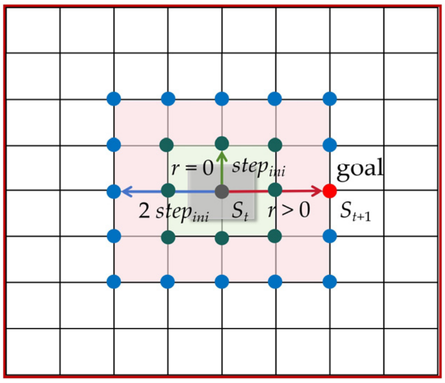


Figure 16. Variable-scale step length strategy.

3. Balance between exploration and exploitation

In addition to effectively setting up the learning environment, ensuring a balance between exploration and exploitation during action selection in the training process is crucial. The aim is to explore all actions and select the most effective one, without being too random to converge or too greedy to get stuck in local optima. To address this issue, this paper introduces an adaptive ε-greedy strategy, which includes an adaptive decay strategy and a balancing strategy, as illustrated in Figure 17. The ε-greedy strategy is a commonly used exploration strategy for agents. As shown in Equation (18), the agent selects a random action with probability ε and the current optimal action with probability $1 - \varepsilon$. The proposed adaptive decay equation is shown in Equation (19), where ε_{high} and ε_{low} represent the maximum and minimum exploration rates, respectively. t and m_T represent the current training episode and the maximum number of episodes, respectively. k and n are decay constants. To bias the agent towards exploration and discovery in the early learning stages and towards exploiting knowledge to select the current optimal action in the later stages, k and n are set to 8 and 4, respectively.

$$\begin{cases} Random\ a, \varepsilon \\ \underset{a}{\operatorname{argmax}} Q(s,a),\ 1-\varepsilon \\ 0 < \varepsilon < 1 \end{cases} \tag{18}$$

$$\varepsilon = \varepsilon_{low} + \left(\varepsilon_{high} - \varepsilon_{low}\right) \times e^{\left(-k \times \left(\frac{t}{m\_T}\right)^n\right)} \tag{19}$$

Through training, this study found that due to the very small value of ε in the later stages, the agent can only choose from a limited action space in a short period within a given state. When the number of ineffective explorations reaches the set limit, the training for that episode ends prematurely, ultimately failing to obtain the optimal action corresponding to the maximum Q value. As shown in Figure 17, after multiple greedy selections, the algorithm only obtained information about two actions. At this point, the training concluded, and the algorithm outputted the suboptimal action a_1, ignoring the optimal action a_3. Therefore, based on the ε-greedy strategy, this study introduces a search-balancing strategy. If certain actions in a given state have not been explored after a certain number of steps (5% of the maximum learning steps), these actions are sequentially selected in the next training step of the current episode.

$$Q^{t+1}(s_t, a_t) \leftarrow Q^t(s_t, a_t) + \alpha\left(r(s_t, a_t, s_{t+1}) + \gamma \max_{a_{t+1}\in A} Q^{\mathrm{t}}(s_{t+1}, a_{t+1}) + \gamma^2 \max_{a_{t+2}\in A} Q^{\mathrm{t}}(s_{t+2}, a_{t+2}) - Q^t(s_t, a_t)\right) \tag{20}$$

In addition to improving action selection strategies, this study adopts a double-step TD strategy for better future prediction and control in the Q-table update process, based on the multi-step TD update concept [48]. The basic version in Equation (13) is modified to Equation (20). As shown in Figure 18, instead of using the predicted reward value r_t from a one-step action (yellow box), the agent uses the predicted reward values r_t and r_{t+1} from two-step actions (red box) to update the Q-table. This strategy provides Q values that are closer to the actual data, with less bias and more stable results, while ensuring time efficiency.

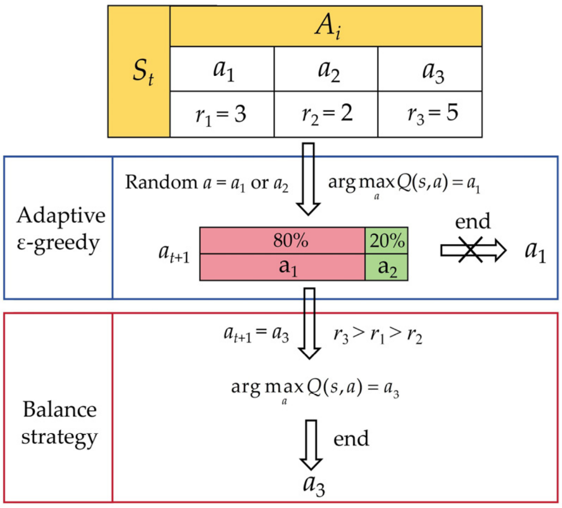


Figure 17. Adaptive ε-greedy strategy.

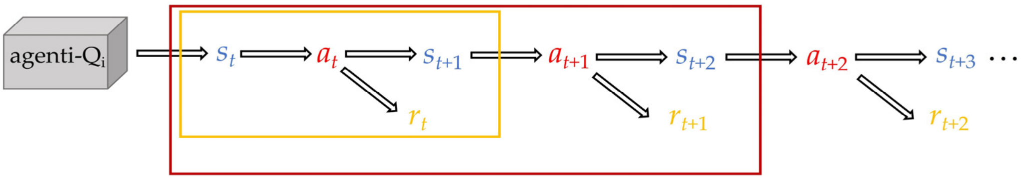


Figure 18. Double-step temporal difference strategy.

3.2.4. Equipment Layout Process

Based on the above improvement strategies, the equipment layout process of AG-MAQL is shown in Figure 19, mainly comprising three nested loops.

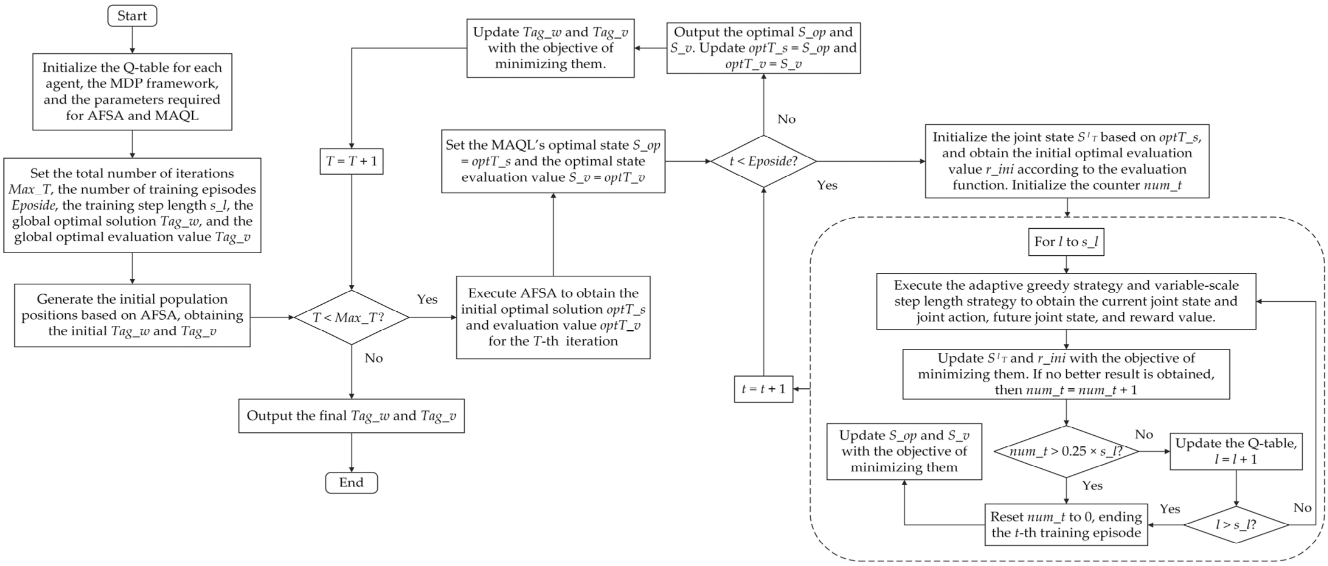

Start
Initialize the Q-table for each agent, the MDP framework, and the parameters required for AFSA and MAQL
Set the total number of iterations Max_T, the number of training episodes Eposide, the training step length s_l, the global optimal solution Tag_w, and the global optimal evaluation value Tag_v
Generate the initial population positions based on AFSA, obtaining the initial Tag_w and Tag_v
T < Max_T?
Yes
No
T = T + 1
Output the final Tag_w and Tag_v
End
Execute AFSA to obtain the initial optimal solution optT_s and evaluation value optT_v for the T-th iteration
Set the MAQL's optimal state S_op = optT_s and the optimal state evaluation value S_v = optT_v
Update Tag_w and Tag_v with the objective of minimizing them.
Output the optimal S_op and S_v. Update optT_s = S_op and optT_v = S_v
t < Eposide?
No
Yes
t = t + 1
Initialize the joint state S¹T based on optT_s, and obtain the initial optimal evaluation value r_ini according to the evaluation function. Initialize the counter num_t
For l to s_l
Execute the adaptive greedy strategy and variable-scale step length strategy to obtain the current joint state and joint action, future joint state, and reward value.
Update S¹T and r_ini with the objective of minimizing them. If no better result is obtained, then num_t = num_t + 1
num_t > 0.25 × s_l?
No
Yes
Update the Q-table, l = l + 1
Update S_op and S_v with the objective of minimizing them
Reset num_t to 0, ending the t-th training episode
l > s_l?
Yes
No

Figure 19. AGMAQL layout flowchart.

3.3. Bidirectional Collaborative Layout Process Based on AGMAQL-ATA

Based on the previously described equipment layout and pipe layout processes, this section proposes the SERLD method based on the collaborative algorithm AGMAQL-ATA, as shown in Figure 20. This method integrates hierarchical layout concepts, an adaptive collaborative weight strategy, and a collaborative evaluation function, and is mainly described as follows:

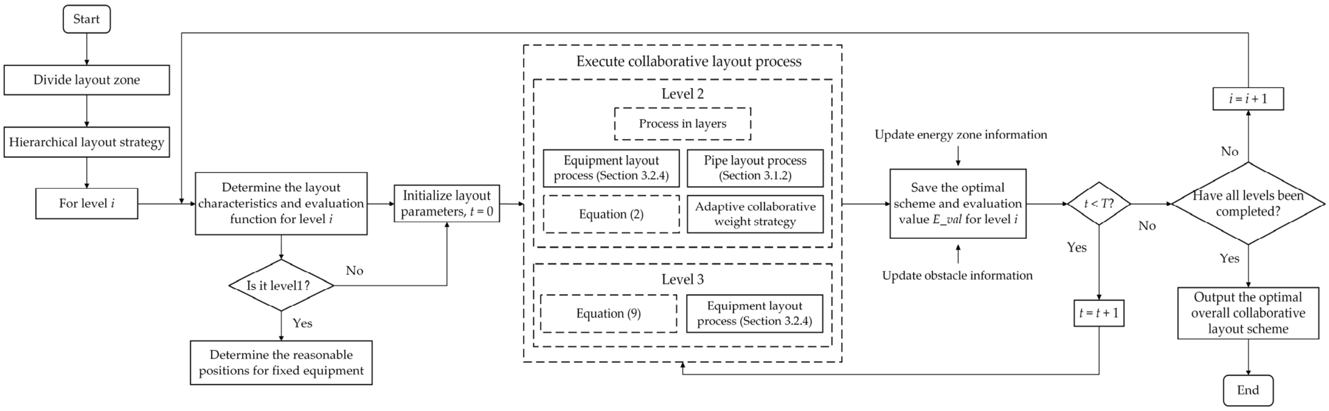

Start
Divide layout zone
Hierarchical layout strategy
For level i
Determine the layout characteristics and evaluation function for level i
Is it level1?
No
Yes
Determine the reasonable positions for fixed equipment
Initialize layout parameters, t = 0
Execute collaborative layout process
Level 2
Process in layers
Equipment layout process (Section 3.2.4)
Pipe layout process (Section 3.1.2)
Equation (2)
Adaptive collaborative weight strategy
Level 3
Equation (9)
Equipment layout process (Section 3.2.4)
Update energy zone information
Save the optimal scheme and evaluation value E_val for level i
Update obstacle information
t < T?
Yes
No
t = t + 1
Have all levels been completed?
No
Yes
i = i + 1
Output the optimal overall collaborative layout scheme
End

Figure 20. The bidirectional collaborative layout flowchart based on AGMAQL-ATA.

At the initial stage of layout, it is necessary to analyze the actual environment of the engine room to be arranged and understand special zones and engineering requirements. After identifying obstacle zones such as restricted zones, passages, and stairways, the feasible layout zones are divided. Our previous research proposed a layout zone allocation procedure considering special zones, torque balance, and functional relevance [6]. In this study, as described in Section 2.2, after determining the feasible layout zone for each piece of equipment, the specific collaborative layout zones are further identified.

In practical engineering, SERLD is influenced by platform structures and the functional nature of the equipment, resulting in different layout priorities [25]. The more equipment there is, the larger the joint state space and action space, leading to higher computational complexity. Therefore, considering the nature of the equipment, engineering practicality, algorithm efficiency, and adaptability, this study proposes a hierarchical layout strategy, as shown in Figure 21, which is divided into three levels: fixed layer, collaborative layer, and equipment layer. The research on SERLD needs to ensure torque balance [6]. Before calculating the layout of each level, the reasonableness of the torque must be judged according to Equation (21), where both sides of the equation represent the torque of the zone on either side of the axis. In this study, the torque error is set within 10%.

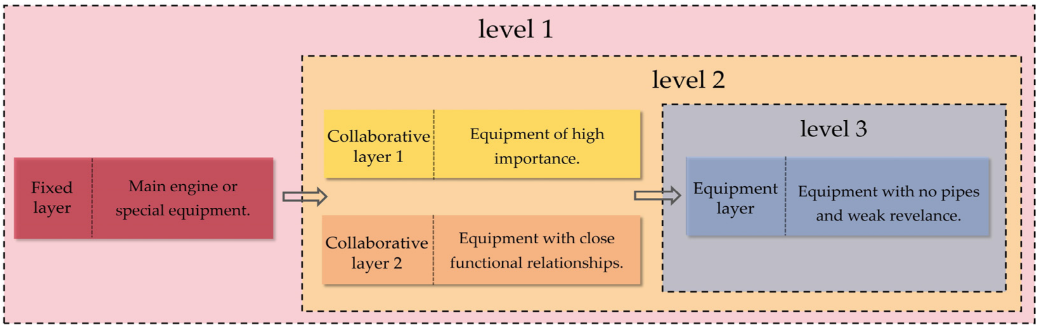

level 1
level 2
level 3
Fixed layer
Main engine or special equipment.
Collaborative layer 1
Equipment of high importance.
Collaborative layer 2
Equipment with close functional relationships.
Equipment layer
Equipment with no pipes and weak revelance.

Figure 21. Hierarchical layout strategy.

$$d_1 \times N_1 \approx d_2 \times N_2 \tag{21}$$

Different levels have different layout approaches. Level 1 has the highest layout priority, typically corresponding to the main engine or specially designated equipment. This level is crucial for adjusting the overall torque balance of the layout. In the initial layout stage, experts need to pre-arrange specific positions to ensure layout rationality and engineering feasibility. Level 2 is the core of the collaborative layout, with layout quality evaluated using Equation (2). This level contains most of the equipment, which needs to be arranged in layers according to its importance. Testing has shown that the layout works optimally when each layer contains fewer than eight pieces of equipment. Level 3 mainly focuses on the remaining equipment without pipe interfaces and with low functional relevance to other equipment, with layout quality evaluated using only Equation (9). In summary, each level maintains interconnectivity, with the optimal layout information from the previous level being incorporated into the calculations for the next level.

In the SERLD problem, the design of the evaluation function directly determines the effectiveness and quality of the collaborative layout. Equipment positioning has a greater impact on SERLD, as pipe layout needs to be based on reasonable equipment positions. Therefore, the evaluation weight of the equipment is higher. To address this, this paper proposes an adaptive weight coefficient strategy to better coordinate the layouts of both layers. As shown in Equation (22), l_1 and l_2 is the weight coefficient in the evaluation Equation (2). Based on the priority of equipment layout, the maximum value of l_1 (l_{high}) is set to 0.8 and the minimum value (l_{low}) to 0.6. t and m_T represent the current and maximum number of iterations, respectively. n is the decay rate, set to 8. Using 100 iterations as an example, the variation curve (red line) in l_1 over time is shown in the three stages (stages a, b, and c in the figure) in Figure 22. Initially, corresponding to stage a (0–10 iterations), to ensure that the equipment can be placed in a reasonable collaborative zone, l_1 is set relatively high. Subsequently, corresponding to stage b (10–20 iterations), to ensure optimization speed, l_2 begins to increase rapidly, reaching the optimal collaborative proportion of 0.6:0.4 within approximately 10% of the iteration cycle. Finally, corresponding to stage c (20+ iterations), in the mid to late stages, balanced collaborative optimization exploration is carried out with a standard proportion.

$$\begin{gathered} l_1 = \mathrm{l}_{low} + \left(\mathrm{l}_{high} - \mathrm{l}_{low}\right) \times \left(1 - \left(\frac{t}{m\_T}\right)^n\right) \\ l_2 = 1 - l_1 \end{gathered} \tag{22}$$

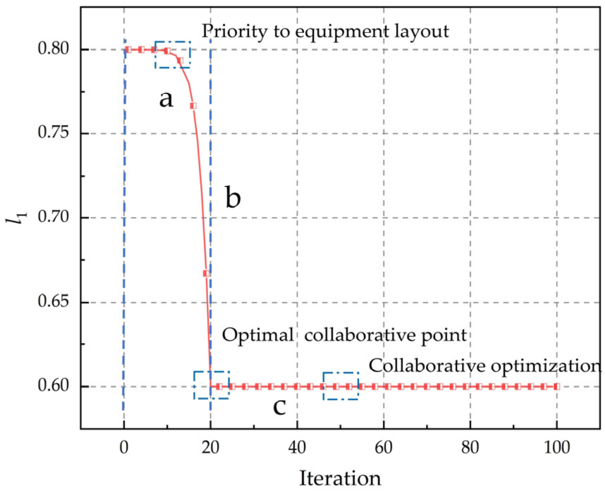


Figure 22. Adaptive collaborative weight strategy.

The above collaborative layout approach aims to achieve joint optimization at both the independent goal layer and the multi-objective layers. Ultimately, according to the engineers' requirements, multiple optimal layout schemes with different focuses can be obtained by adjusting the weight coefficients.

4. Verification Analysis and Discussion Based on Practical Case

4.1. Case Information and Experimental Conditions

To verify the feasibility of the collaborative layout method proposed in this paper, this experiment uses a complex and representative actual ship engine room from Reference [6] as the test case. This study is validated in terms of encoding, algorithms, and layout strategies. The experimental environment includes a Windows 11 operating system, Python 3.12 simulation tools, and a 12th Gen Intel(R) Core(TM) i7-12700H processor.

Figure 23 shows the original manually arranged engine room in this case, which includes 21 pieces of equipment, such as a nuclear reactor. The model after envelope processing is shown in Figure 24. As described in Section 2.2, this case can be divided into two main feasible layout zones. The collaborative layout zone for each piece of equipment is initially set by experts, with each piece roughly positioned at the center of its respective zone. According to the hierarchical layout strategy, Figure 24 shows the information for different layout levels. Level 2 is divided into two layers due to the limitation on the amount of equipment. There are 15 pipes in total, with interface coordinates based on the relative positions of the equipment center-points. E denotes equipment numbers, and S-pipe and B-pipe represent single and branch pipes, respectively. According to Section 3.3, the central positions of level 1 equipment E1 and E2 are set at (33.5, 53, 16) and (15, 6, 30.5), respectively. It should be noted that level 1 is set considering the torque balance of all equipment. However, to ensure the rationality of torque distribution during the evaluation of the level 2-1 layout, level 1 is temporarily excluded from torque calculations. Additionally, the relevance coefficients between all equipment can be found in the original literature. Equipment E3, E4, and E21, which are on the bottom platform of the engine room along with the nuclear reactor E1, cannot have their *z*-coordinates changed, and the height of other equipment cannot be lower than that of the platform.

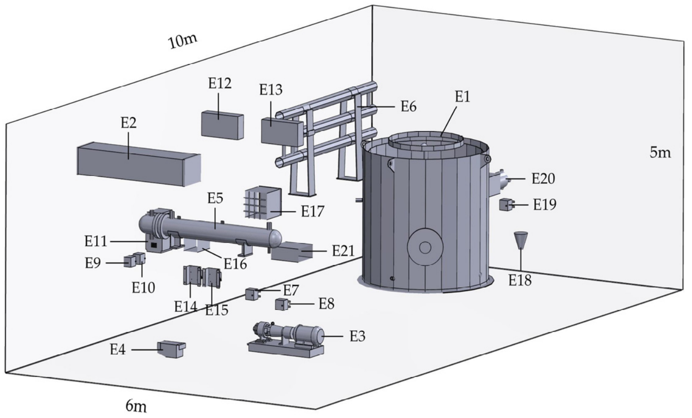
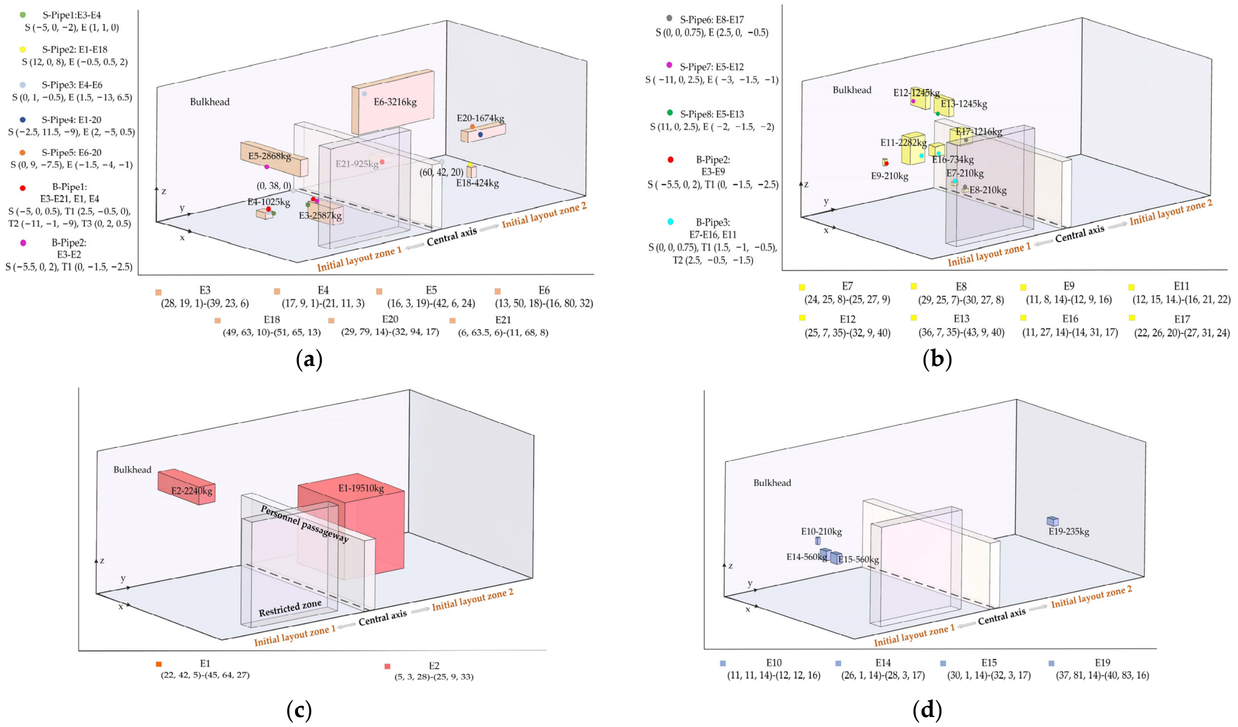


Figure 23. Actual layout of the original engine room [6].



(**a**) (**b**) (**c**) (**d**)

Figure 24. Layout hierarchy and parameterized information: (**a**) level 2-1; (**b**) level 2-2; (**c**) level 1; (**d**) level 3.

4.2. Experimental Setup

To reasonably verify the feasibility of the novel SERLD method proposed in this study, comparative validation is conducted from two main perspectives: layout algorithms and layout strategies. The layout algorithms include the ATAFSA and AGMAQL, while the layout strategies include hierarchical strategy and adaptive collaborative weight strategy.

4.2.1. Experimental Comparison of Layout Algorithms

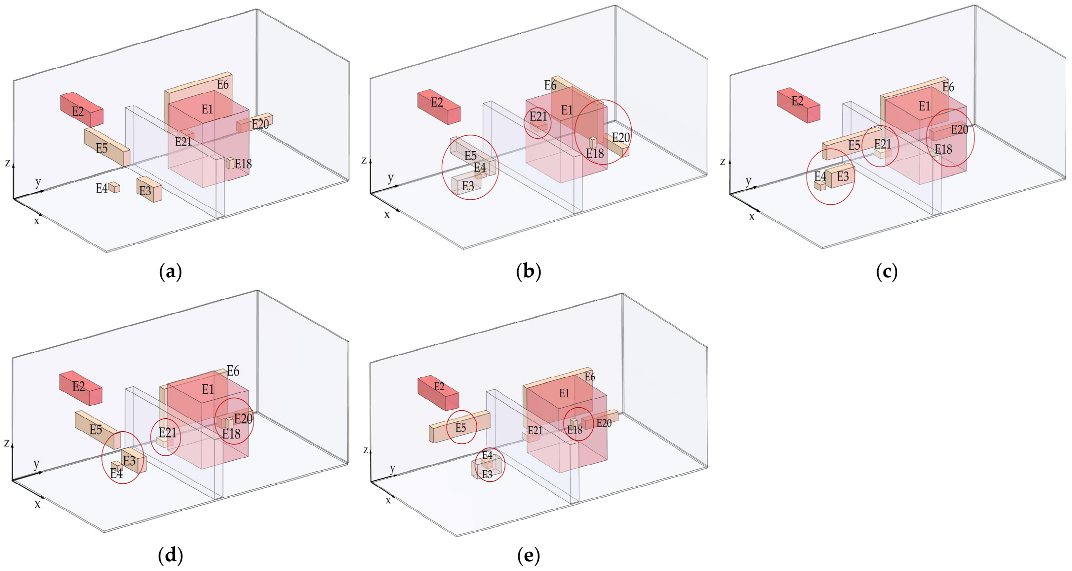
In terms of layout algorithms, the validation of AGMAQL used for equipment layout and the ATAFSA used for pipe layout is conducted separately. The specific experimental setup is as follows:

1. For the equipment layout algorithm AGMAQL, this experiment approaches the comparison from the perspectives of reinforcement learning and optimization algorithms: On one hand, AGMAQL is compared with the HMSAFSA from previous SERLD research [6] to verify its applicability relative to optimization algorithms. Previous

research focused on finding the optimal layout through equipment translation actions and demonstrated the optimization performance of the HMSAFSA through comparisons with various optimization algorithms. On the other hand, AGMAQL is compared with the basic MAQL algorithm and Wolf-PHC [30] to verify its effectiveness relative to other MARL algorithms. Wolf-PHC is a leading algorithm in the field of facility layout. For this algorithm, each agent also maintains an independent Q-table and can adjust learning parameters and strategies based on its performance. Finally, AGMAQL is compared with manual layout schemes to validate its engineering practicality.

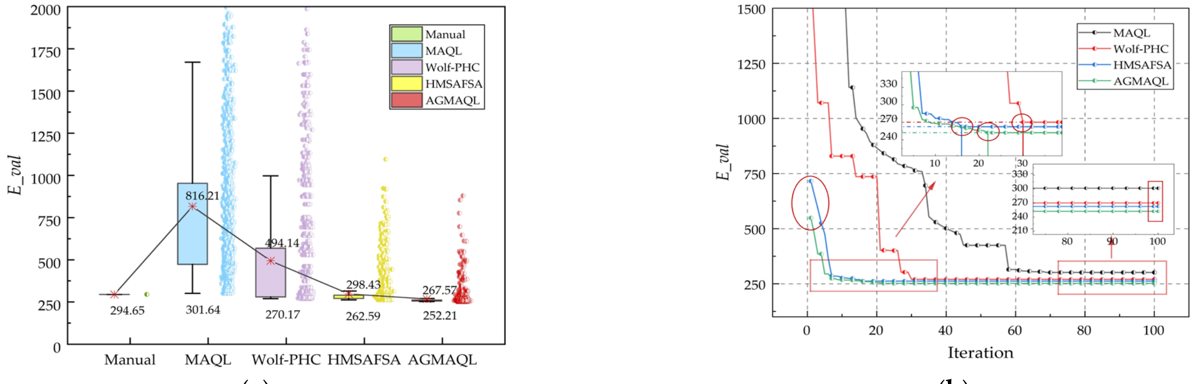
2. For the pipe layout ATAFSA, this experiment primarily focuses on the underlying encoding and algorithm improvements, comparing it with the leading research. The comparison includes the following two aspects: On one hand, the ATAFSA is compared with the leading vector encoding method [4] and the Manhattan trajectory-based encoding method [6], both of which have been proven effective through extensive experiments. On the other hand, the ATAFSA is compared with the original AFSA, and the HMSAFSA and DDECS algorithm from the literature, which are leading optimization algorithms in the SPLP field. Through these comparisons, the optimization speed, accuracy, and stability of the ATAFSA are validated.

To ensure the fairness of the experiment, algorithm parameters are set according to the optimal ranges specified in the original literature. The parameter settings for the HMSAFSA, AFSA, and ATAFSA remain consistent. It should be noted that the feasible variation space for vector points in this study is significantly reduced compared to previous research, necessitating parameter adjustments. The aim is to ensure that the movement variation value of the population in each iteration is about 10% of the maximum feasible range [5]. After verification, the adjustment principle is as follows: when the feasible variation space of vector points is reduced to x% of the original, the core action parameters S_V and S_S of the algorithm are also reduced to x% of their original values. In this study, the feasible variation space for pipe layout is reduced by approximately 80%, and for equipment layout by approximately 70%. Furthermore, this study uses the core and representative levels 1 and 2-1 as validation cases for the algorithm. During pipe layout, the equipment positions remain unchanged. The number of tests is 20, with a population size of 50 and 100 iterations for each test. The final test results are shown in Figures 25–28 and Tables 3 and 4, where E_val represents the layout evaluation value. The suffixes T and A represent the encoding methods proposed in Reference [4] and this study, respectively.



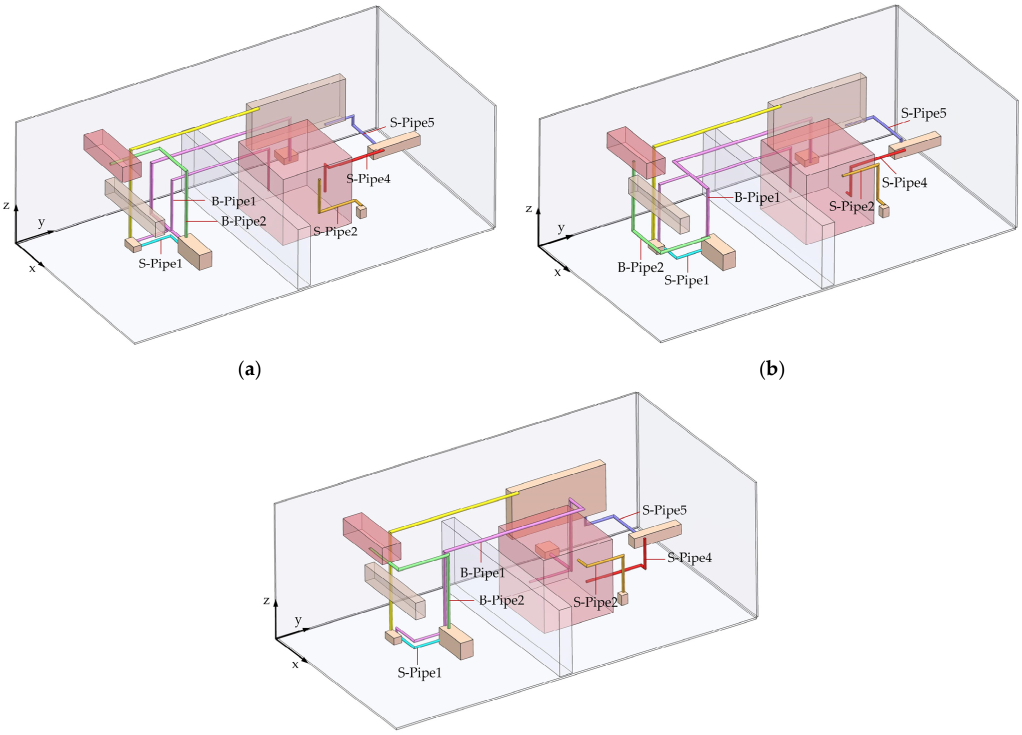
(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 25. Comparison of optimal layout schemes for various algorithms (equipment layout): (**a**) manual experience; (**b**) MAQL; (**c**) Wolf-PHC; (**d**) HMSAFSA; (**e**) AGMAQL.



(**a**) (**b**)

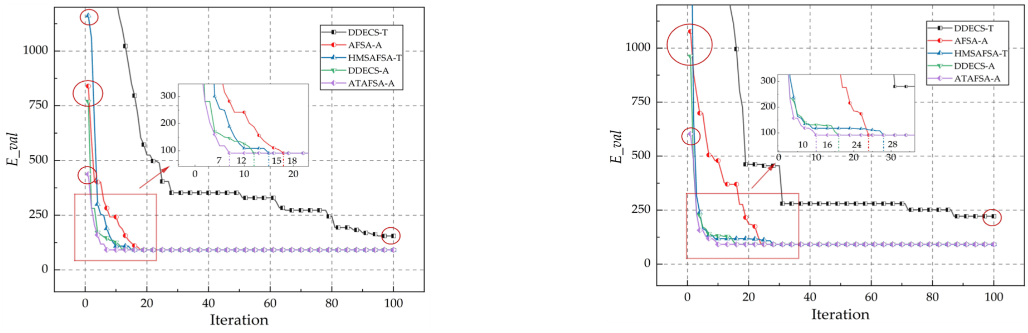
Figure 26. Comparison of optimization performance for various algorithms (equipment layout): (**a**) all test results; (**b**) optimal test results.



(**a**) (**b**)

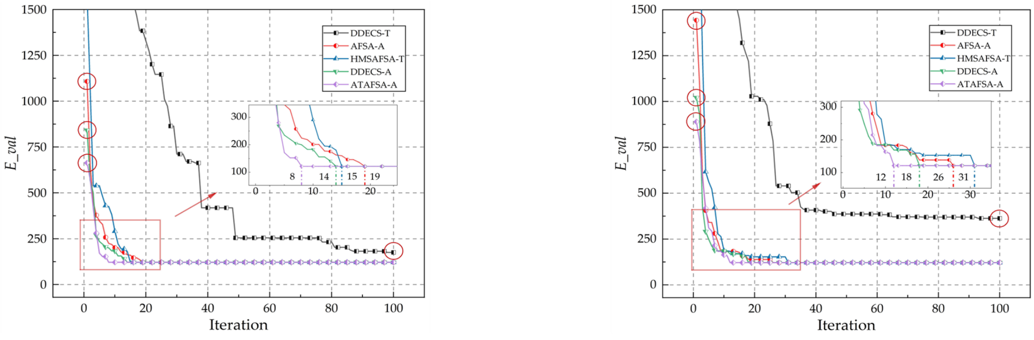
(**c**)

Figure 27. Optimal layout schemes for various algorithms (pipe layout): (**a**) scheme 1; (**b**) scheme 2; (**c**) scheme 3.



(**a**) (**b**)

Figure 28. *Cont.*



(**c**) (**d**)

Figure 28. Comparison of optimization performance for various algorithms (pipe layout): (**a**) optimal test results (S-pipe3); (**b**) worst test results (S-pipe3); (**c**) optimal test results (B-pipe1_main); (**d**) worst test results (B-pipe1_main).

Table 3. Detailed comparison of evaluation data (equipment layout).

	Optimal *E_val*	**Deviation**	**Average *E_val***	**Deviation**
Manual	294.65	14.4%	None	None
MAQL	301.64	16.4%	816.21	67.2%
Wolf-PHC	270.17	6.6%	494.14	45.9%
HMSAFSA	262.59	4.0%	298.43	10.3%
AGMAQL	252.21	None	267.57	None

Table 4. Detailed comparison of optimization performance data (pipe layout).

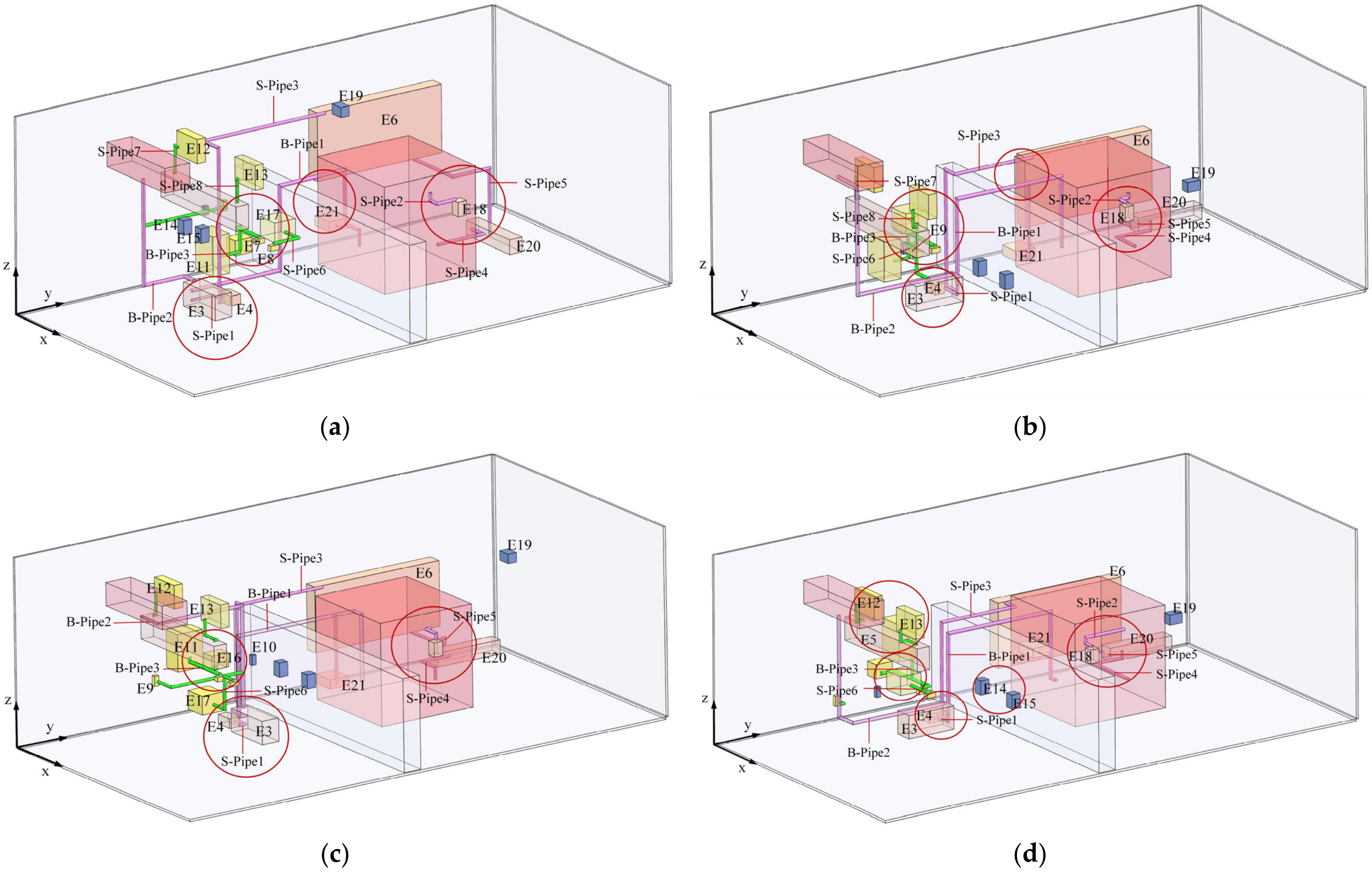
	Convergence Iterations (S-pipe3)				**Convergence Iterations (B-pipe1_Main)**			
	Best	**Worst**	**Average**	**Deviation**	**Best**	**Worst**	**Average**	**Deviation**
DDECS-T	100+	100+	100+	91.9%+	100+	100+	100+	90.8%+
HMSAFSA-T	15	28	21.5	62.3%	15	31	23.9	61.5%
AFSA-A	18	24	20.7	60.9%	19	26	22.3	58.7%
DDECS-A	12	16	13.6	40.4%	14	18	15.7	41.4%
ATAFSA-A	7	10	8.1	None	8	12	9.2	None

4.2.2. Experimental Comparison of Layout Strategies

To verify the effectiveness of the layout strategies, this study compares the proposed strategies with ideas from other literature and incorporates the concept of ablation experiments to validate the effectiveness of the two proposed collaborative strategies. The specific experimental setup is as follows:

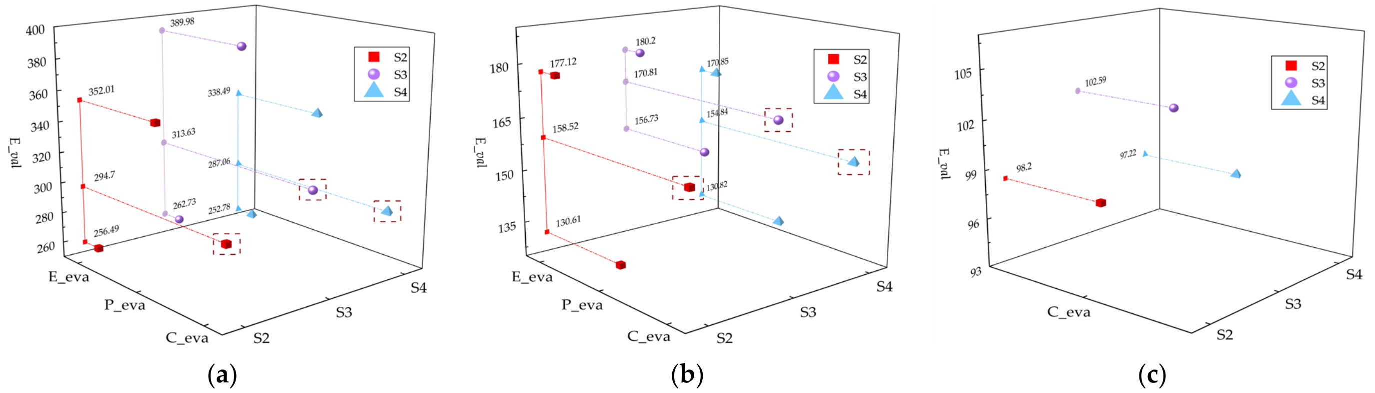
1. Strategy 1: Validate the effectiveness of the hierarchical strategy. This strategy follows the conventional layout approach in SERLD research, where no hierarchical processing is performed during collaborative layout. All equipment and pipes are processed at the same level, finding the optimal solution through function constraints. This method is based on the idea in Reference [22].
2. Strategy 2: Validate the effectiveness of the adaptive collaborative weight strategy. This follows the traditional layout approach [29], where the evaluation weights for equipment and pipes remain constant throughout the collaborative layout process, specifically $l_1 = 0.6$ and $l_2 = 0.4$.
3. Strategy 3: Compare with the most advanced SERLD method from the latest reference [6]; this method also has similar hierarchical and collaborative guidance strategies, but the equipment does not have rotational actions.
4. Strategy 4: The layout strategy proposed in this paper.

All comparative tests for the strategies are conducted under a unified environment, including the collaborative algorithm AGMAQL-ATA, evaluation methods, and weight coefficients. Each strategy underwent 20 sets of tests, with each set having 100 iterations. The final test results are shown in Figures 29–31 and Table 5, where S represents the abbreviation for strategy. E_eva, P_eva, and C_eva represent the optimization objectives for the layers of equipment, pipes, and collaboration, respectively.



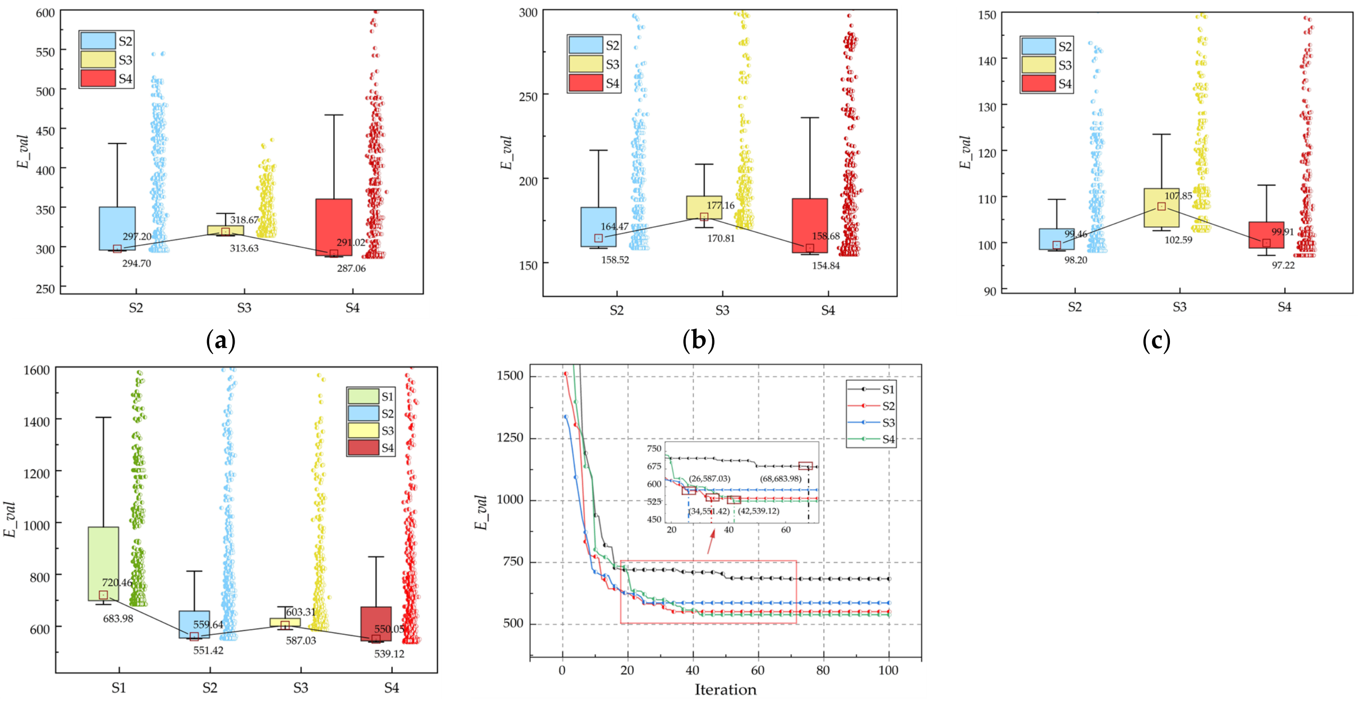
(a) (b) (c) (d)

Figure 29. Comparison of optimal layout schemes for various strategies (collaborative layout): (**a**) S1; (**b**) S2; (**c**) S3; (**d**) S4.



(a) (b) (c)

Figure 30. Comparison of optimal optimization results for various objectives at different levels: (**a**) level 2-1; (**b**) level 2-2; (**c**) level 3.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 31. Comparison of optimization performance for various strategies (collaborative layout): (**a**) level 2-1; (**b**) level 2-2; (**c**) level 3; (**d**) comprehensive test results; (**e**) optimal test results.

Table 5. Comprehensive comparison of evaluation data (collaborative layout).

	Level 2-1		**Level 2-2**		**Level 3**		**Overall *E_val***	**Deviation**	**Median *E_val***	**Deviation**
	Optimal *E_val*	**Median *E_val***	**Optimal *E_val***	**Median *E_val***	**Optimal *E_val***	**Median *E_val***				
S1	None	None	None	None	None	None	683.98	21.2%	720.46	23.7%
S2	294.70	297.20	158.52	164.47	98.20	99.46	551.42	2.2%	559.64	1.7%
S3	313.63	318.67	170.81	177.16	102.59	107.85	587.03	8.2%	603.31	8.8%
S4	287.06	291.02	154.84	158.68	97.22	99.91	539.12	None	550.05	None

4.3. Analysis and Discussion of Test Results

4.3.1. Equipment Layout Aspects

In terms of equipment layout, the following test results were obtained through extensive comparative testing with manual experience, leading optimization algorithms, and leading reinforcement learning algorithms: the optimal layout scheme (Figure 25), overall optimization data, and the optimal optimization data (Figure 26 and Table 3).

All algorithms considered aspects such as balance, relevance, and compliance, among others, resulting in the optimal layout scheme depicted in Figure 25, with major changes marked. According to Figure 26 and Table 3, except for MAQL, all algorithms showed significant improvements compared to manual experience. The AGMAQL proposed in this paper demonstrated the most significant layout enhancement, validating its feasibility and engineering practicality. The optimal and average *E_val* for MAQL differed by 16.4% and 67.2%, respectively, compared to our algorithm, showing lower optimization stability and a tendency to fall into local optima, resulting in a higher overall *E_val*. Wolf-PHC shows considerable improvements in optimization efficiency and stability compared to MAQL, but it still lags behind AGMAQL and has poor robustness. As seen, compared to reinforcement learning algorithms in the literature, AGMAQL maintained its optimization performance even when dealing with multiple agents and large state spaces, ensuring stable optimization of multiple variables. In addition to the comparison of reinforcement learning algorithms, this experiment also compared AGMAQL with the leading optimization algorithm, the HMSAFSA. As shown in Figure 26b, the HMSAFSA demonstrated stability and efficiency when applied to equipment layout. However, Figure 26a reveals that although optimization algorithms are highly efficient, their optimization is somewhat limited due to their iterative logic and parameter factors. The optimal and average *E_val* of the HMSAFSA differ from those of AGMAQL by 4.0% and 10.3%, respectively. Compared to optimization algorithms, reinforcement learning offers flexible fine optimization capabilities, is less affected by parameters, and can continuously adjust the optimization extent of equipment according to environmental changes, making it more targeted. The test results proved that AGMAQL significantly improves the layout upper limit, and finds a globally better scheme, confirming its applicability and engineering practicality in equipment layout problems.

4.3.2. Pipe Layout Aspects

In terms of pipe layout, comparative tests are conducted with currently widely used algorithms and encoding methods in this field, based on the same equipment position scheme. This resulted in three different types of optimal layout schemes (Figure 27). Subsequently, the optimal and worst optimization performances of various algorithms are compared based on different complex pipe cases (Figure 28 and Table 4).

Based on 20 sets of comparative tests, the following conclusions can be drawn: (1) Both the literature algorithms and the proposed algorithm can achieve better layout results, validating the feasibility and flexibility of applying optimization algorithms to pipe layout. This paper selects three representative better layout schemes, as shown in Figure 27. These algorithms can adjust objective weights according to the engineer's requirements to achieve different optimal layouts. For example, Figure 27a,b focus more on the bends and the length of the pipes, while Figure 27c emphasizes the utilization of the energy zone. (2) In terms of encoding validation, as shown in Figure 28, the tests on two complex single pipe cases (S-Pipe3 and B-Pipe1_main) indicate that the optimal and worst optimization performances of ATAFSA-A, DDECS-A, and AFSA-A are superior to those of DDECS-T and HMSAFSA-T. Specifically, as indicated by the red circles and squares in the figure, they can find a relatively better region in the initial stage and quickly converge to the optimal *E_val* within 7-26 iterations. Table 4 shows that the optimization results of DDECS-A and DDECS-T, which use different encoding methods but the same algorithm, differ by more than 50%, further proving the efficiency of the proposed encoding method. (3) In terms of algorithm validation, a comparison is made between the proposed algorithm ATAFSA-A, the literature algorithm DDECS-A, and the original AFSA-A, all using the same encoding method. As shown in the red annotations in Figure 28 and Table 4, the optimization efficiency of DDECS-A and AFSA-A differs from ATAFSA-A by approximately 40% and 60%, respectively, with ATAFSA-A having the best initial solution and requiring only about eight iterations to find the optimal solution. Additionally, it is noteworthy that the ATAFSA is an improvement on our previous research, the HMSAFSA. The tests revealed that the ATAFSA maintains stable efficiency even in the worst optimization scenarios, whereas the HMSAFSA is less stable, with a significant gap between its best and worst optimization results. This validates the feasibility of the parallel optimization strategy proposed in this paper. Through the above tests, the efficiency and stability of the ATAFSA were verified.

4.3.3. Collaborative Layout Aspects

In terms of collaborative layout, extensive and detailed comparative tests were conducted using ablation concepts and strategy ideas from the literature to validate the feasibility of the proposed layout strategy. The following test results were obtained: the optimal layout schemes (Figure 29), the optimal optimization results for various objectives at each layout level (Figure 30), and the comprehensive collaborative test results (Figure 31 and Table 5).

Based on comparative tests from different perspectives, the following experimental conclusions can be drawn: (1) According to Figure 29 and Table 5, the optimal layout scheme obtained by the proposed strategy features the shortest pipe length, the fewest bends, a large proportion of the energy zone, balanced overall torque, strong layout relevance, and attention to safety and engineering constraints. Major changes in other schemes are marked in the figure, and their collaborative values *E_val* differ from the proposed strategy by 21.2%, 2.2%, and 8.2%, respectively. (2) According to Figures 30 and 31, by comparing the optimization objectives E_eva, P_eva, and C_eva at each level, a consistent pattern is observed in levels 2-1 and 2-2: the adaptive collaborative weight strategy proposed in this paper results in higher *E_val* in the early optimization stages, leading to larger errors. However, by comparing the median and optimal data across the entire test, it is evident that S2 and S3 only achieve local optimal results, which are inferior to the proposed strategy. For level 3, since it only involves equipment objectives, the results of S2 and the proposed strategy are almost identical, while S3 still performs poorly due to not considering more collaborative actions and zones. (3) According to Figure 31d,e in 20 sets of tests, S1 performed the worst. This indicates that without adopting a hierarchical approach, the significant increase in optimization variables leads to a decline in algorithm performance. Ultimately, the optimal *E_val* and median for S1 differ by more than 20% from the proposed strategy. Based on all the data included in Figure 31, the test results for S2 and S3 are consistent with conclusion (2): although they optimize quickly, they have lower optimization upper limits, making it difficult to find globally better solutions, and S3 is prone to local optima. In summary, the collaborative layout strategy proposed in this paper shows significant improvements compared to other literature strategies, further proving the practicality and rationality of this research. It can efficiently provide designers with the most valuable reference schemes.

5. Conclusions

To address the research deficiencies in SERLD, this paper proposes a novel layout method that includes optimization strategies for both independent and overall levels. This method effectively combines reinforcement learning and heuristic algorithms, significantly enhancing the upper limit of intelligent layout results in SERLD. The main contributions are as follows:

1. In terms of equipment layout, to address the multi-variable optimization challenges of algorithms, this paper creatively applies the multi-agent reinforcement learning algorithm MAQL to SERLD research, proposing an improved AGMAQL algorithm. This algorithm focuses optimization efforts on easily controllable equipment. AGMAQL features a reasonable MDP framework, improved state, action, and learning strategies. Compared with other reinforcement learning and optimization algorithms, AGMAQL achieved over a 4.0% improvement in layout effectiveness at the equipment level, validating its efficiency and rationality.
2. In terms of pipe layout, to address the instability of pipe optimization in collaborative layout problems, this paper proposes a powerful adaptive trajectory-based encoding method and an improved algorithm, the ATAFSA. This algorithm integrates parameter-adaptive strategy, scouting optimization strategy, and parallel optimization strategy. Through testing and comparison at both the encoding and algorithm levels, the ATAFSA achieved an over 40.4% improvement in optimization efficiency at the pipe level, validating its stability and suitability for collaborative applications.
3. In terms of collaborative layout, to overcome the deficiencies in traditional independent layout strategies, this paper considers actual engineering specifications and multi-level objectives and constraints, proposing a SERLD method that includes an adaptive collaborative weight strategy, a hierarchical layout strategy, and a more comprehensive collaborative evaluation function. While simplifying the problem, these strategies effectively achieve collaborative optimization of equipment and pipes. Finally, based on a practical engine room case and through comparison with multiple

literature strategies, AGMAQL-ATA achieved an over 2.2% improvement in layout effectiveness at the collaborative level, validating the feasibility and engineering practicality of the proposed strategies.

The proposed SERLD method addresses previous research gaps, provides new insights into SERLD research, significantly improves ship design efficiency, and can flexibly provide engineers with reference schemes from different perspectives. However, the current research exploration mainly focuses on the geometric aspects and typical engine room environments, without addressing more practical factors such as manufacturing, structure, gravity, fluid dynamics, and stability. In the future, collaborative research on SERLD will further consider more practical engineering issues in shipbuilding, such as design for hull structure and curved surface; design for optimal manufacturing; and design for repair and disassembly, stability, and ship weight optimization. By rationally arranging the engine room, the intelligence, efficiency, and stability of ship design can be enhanced.

Author Contributions: Conceptualization, H.Z.; methodology, H.Z.; software, H.Z. and L.T.; validation, H.Z. and L.T.; formal analysis, H.Z. and Z.S.; investigation, Y.Y., Y.H., and Z.Y.; resources, Y.Y.; data curation, H.Z., Z.S., and Y.H.; writing—original draft preparation, H.Z.; writing—review and editing, Y.Y. and Z.S.; visualization, H.Z.; supervision, Y.Y.; project administration, Y.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (Grant No. 51409042).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data related to the actual ship engine room cannot be disclosed due to confidentiality. All other data are contained within this article.

Acknowledgments: The authors appreciate shipowners for providing actual engine room layout information and specification documents.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Anjos, M.F.; Vieira, M.V. Mathematical optimization approaches for facility layout problems: The state-of-the-art and future research directions. *Eur. J. Oper. Res.* **2017**, *261*, 1–16. [CrossRef]
2. Dong, Z.; Lin, Y. A particle swarm optimization based approach for ship pipe route design. *Int. Shipbuild. Prog.* **2017**, *63*, 59–84. [CrossRef]
3. Bian, X.; Lin, Y.; Dong, Z.X. Auto-routing methods for complex ship pipe route design. *J. Ship Prod. Des.* **2022**, *38*, 100–114. [CrossRef]
4. Lin, Y.; Bian, X.; Dong, Z.R. A discrete hybrid algorithm based on Differential Evolution and Cuckoo Search for optimizing the layout of ship pipe route. *Ocean Eng.* **2022**, *261*, 112164. [CrossRef]
5. Zhang, H.; Yang, M.; Yang, Y.; Liu, H.; Lin, Y. Collaborative Layout Optimization for Ship Pipes Based on Spatial Vector Coding Technique. *IEEE Access* **2023**, *11*, 116762–116785. [CrossRef]
6. Zhang, H.; Yu, Y.; Zhang, Q.; Yang, Y.; Liu, H.; Lin, Y. A bidirectional collaborative method based on an improved artificial fish swarm algorithm for ship pipe and equipment layout design. *Ocean Eng.* **2024**, *296*, 117045. [CrossRef]
7. Dijkstra, E.W. A note on two problems in connexion with graphs. In *Edsger Wybe Dijkstra: His Life, Work, and Legacy*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 287–290. [CrossRef]
8. Lee, C.Y. An algorithm for path connections and its applications. *IRE Trans. Electron. Computers.* **1961**, *EC-10*, 346–365. [CrossRef]
9. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [CrossRef]
10. Dong, Z.; Bian, X. Ship pipe route design using improved A* algorithm and genetic algorithm. *IEEE Access* **2020**, *8*, 153273–153296. [CrossRef]
11. Dong, Z.; Bian, X.; Zhao, S. Ship pipe route design using improved multi-objective ant colony optimization. *Ocean Eng.* **2022**, *258*, 111789. [CrossRef]
12. Niu, W.; Sui, H.; Niu, Y.; Cai, K.; Gao, W. Ship Pipe Routing Design Using NSGA-II and Coevolutionary Algorithm. *Math. Probl. Eng.* **2016**, *2016*, 7912863. [CrossRef]

13. Lin, Y.; Zhang, Q. A multi-objective cooperative particle swarm optimization based on hybrid dimensions for ship pipe route design. *Ocean Eng.* **2023**, *280*, 114772. [CrossRef]
14. Ha, J.; Roh, M.I.; Kim, K.S.; Kim, J.H. Method for pipe routing using the expert system and the heuristic pathfinding algorithm in shipbuilding. *Int. J. Nav. Archit. Ocean Eng.* **2023**, *15*, 100533. [CrossRef]
15. Kim, Y.; Lee, K.; Nam, B.; Han, Y. Application of reinforcement learning based on curriculum learning for the pipe auto-routing of ships. *J. Comput. Des. Eng.* **2023**, *10*, 318–328. [CrossRef]
16. Dong, Z.; Luo, W. Ship pipe route design based on NSGA-III and multi-population parallel evolution. *Ocean Eng.* **2024**, *293*, 116666. [CrossRef]
17. Gan, Q. A logistics distribution route optimization model based on hybrid intelligent algorithm and its application. *Ann. Oper. Res.* **2022**, 1–13. [CrossRef]
18. Li, F.F.; Du, Y.; Jia, K.J. Path planning and smoothing of mobile robot based on improved artificial fish swarm algorithm. *Sci. Rep.* **2022**, *12*, 659. [CrossRef] [PubMed]
19. Zhao, L.; Bai, Y.; Wang, F.; Bai, J. Path planning for autonomous surface vessels based on improved artificial fish swarm algorithm: A further study. *Ships Offshore Struct.* **2023**, *18*, 1325–1337. [CrossRef]
20. Luo, X.; Yang, Y.; Ge, Z.; Wen, X.; Guan, F. Maintainability-based facility layout optimum design of ship cabin. *Int. J. Prod. Res.* **2015**, *53*, 677–694. [CrossRef]
21. Lee, K.Y.; Roh, M.I.; Jeong, H.S. An improved genetic algorithm for multi-floor facility layout problems having inner structure walls and passages. *Comput. Educ.* **2015**, *32*, 879–899. [CrossRef]
22. Besbes, M.; Zolghadri, M.; Costa Affonso, R.; Masmoudi, F.; Haddar, M. A methodology for solving facility layout problem considering barriers: Genetic algorithm coupled with A* search. *J. Intell. Manuf.* **2020**, *31*, 615–640. [CrossRef]
23. Lee, B.C.; Choi, Y.; Chung, H. Firefighting equipment arrangement optimization for an offshore platform considering travel distances. *J. Mar. Sci. Eng.* **2021**, *9*, 503. [CrossRef]
24. Mallam, S.C.; Lundh, M.; MacKinnon, S.N. Integrating Human Factors & Ergonomics in large-scale engineering projects: Investigating a practical approach for ship design. *Int. J. Ind. Ergon.* **2015**, *50*, 62–72. [CrossRef]
25. Meng, X.; Sun, H.; Kang, J. Equipment layout optimization based on human reliability analysis of cabin environment. *J. Mar. Sci. Eng.* **2021**, *9*, 1263. [CrossRef]
26. Zhang, Q.; Zhang, H.; Lin, Y.; Yang, Y.; Liu, H. A Methodology for Ship Cabin Equipment Layout Considering Human Factor Reliability Optimization. *J. Ship Prod. Des.* **2024**, 1–12. [CrossRef]
27. Jang, W.; Lin, Y.; Chen, M.; Yu, Y. An optimization approach based on particle swarm optimization and ant colony optimization for arrangement of marine engine room. *J. Shanghai Jiaotong Univ.* **2014**, *48*, 502. [CrossRef]
28. Haris, N.; Sohn, J.M.; Prabowo, A.R. Layout optimization for safety evaluation on LNG-fueled ship under an accidental fuel release using mixed-integer nonlinear programming. *Int. J. Nav. Archit. Ocean Eng.* **2022**, *14*, 100443. [CrossRef]
29. Gunawan, G.; Utomo, A.S.A.; Hamada, K.; Ouchi, K.; Yamamoto, H.; Sueshige, Y. Optimization of module arrangement in ship engine room. *J. Ship Prod. Des.* **2021**, *37*, 54–66. [CrossRef]
30. Zhang, Q.; Lin, Y. Integrating multi-agent reinforcement learning and 3D A* search for facility layout problem considering connector-assembly. *J. Intell. Manuf.* **2023**, 1–26. [CrossRef]
31. Bengio, Y.; Lodi, A.; Prouvost, A. Machine learning for combinatorial optimization: A methodological tour d'horizon. *Eur. J. Oper. Res.* **2021**, *290*, 405–421. [CrossRef]
32. Mazyavkina, N.; Sviridov, S.; Ivanov, S.; Burnaev, E. Reinforcement learning for combinatorial optimization: A survey. *Comput. Educ.* **2021**, *134*, 105400. [CrossRef]
33. Yang, Z.; Liu, Y.; Chen, Y.; Al-Dhahir, N. Cache-aided NOMA mobile edge computing: A reinforcement learning approach. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 6899–6915. [CrossRef]
34. Yin, Y.; Guo, Y.; Su, Q.; Wang, Z. Task allocation of multiple unmanned aerial vehicles based on deep transfer reinforcement learning. *Drones.* **2022**, *6*, 215. [CrossRef]
35. Wu, Q.; Lin, R.; Ren, Z. Distributed multirobot path planning based on MRDWA-MADDPG. *IEEE Sens. J.* **2023**, *23*, 25420–25432. [CrossRef]
36. Adeogun, R.; Berardinelli, G. Multi-agent dynamic resource allocation in 6G in-X subnetworks with limited sensing information. *Sensors* **2022**, *22*, 5062. [CrossRef] [PubMed]
37. Zhou, C.; Stephen, A.; Tan, K.C.; Chew, E.P.; Lee, L.H. Multiagent Q-Learning Approach for the Recharging Scheduling of Electric Automated Guided Vehicles in Container Terminals. *Transp. Sci.* **2024**, *58*, 664–683. [CrossRef]
38. Wang, T.; Cao, J.; Hussain, A. Adaptive Traffic Signal Control for large-scale scenario with Cooperative Group-based Multi-agent reinforcement learning. *Transp. Res. Part C Emerg. Technol.* **2021**, *125*, 103046. [CrossRef]
39. Dixit, V.; Verma, P.; Raj, P. Leveraging tacit knowledge for shipyard facility layout selection using fuzzy set theory. *Expert Syst. Appl.* **2020**, *158*, 113423. [CrossRef]
40. Asmara, A. Pipe Routing Framework for Detailed Ship Design. Ph.D. Thesis, TUD Technische Universiteit Delft, Delft, The Netherlands, 2013; p. 155. [CrossRef]
41. Li, X. A New Intelligent Optimization Method-Artificial Fish School Algorithm. Ph.D. Thesis, Zhejiang University, Hangzhou, China, 2003. [CrossRef]
42. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT press: Cambridge, MA, USA, 2018. [CrossRef]

43. Watkins, C.J.; Dayan, P. Q-learning. *Int. J. Mach. Learn. Cybern.* **1992**, *8*, 279–292. [CrossRef]
44. Jang, B.; Kim, M.; Harerimana, G.; Kim, J.W. Q-learning algorithms: A comprehensive classification and applications. *IEEE Access* **2019**, *7*, 133653–133667. [CrossRef]
45. Kar, S.; Moura, J.M.; Poor, H.V. Qd-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus. *arXiv preprint* **2012**, arXiv:1205.0047. [CrossRef]
46. Awheda, M.D.; Schwartz, H.M. Exponential moving average Q-learning algorithm. In *2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*; IEEE: New York, NY, USA, 2013; pp. 31–38. [CrossRef]
47. Arta, M.N.; Gunawan, G.; Yanuar, Y. Part arrangement optimization in ship engine room based on the genetic algorithm. *AIP Conf. Proc.* **2020**, *2255*, 020009. [CrossRef]
48. De Asis, K.; Hernandez-Garcia, J.; Holland, G.; Sutton, R. Multi-step reinforcement learning: A unifying algorithm. In Proceedings of the AAAI Conference on Artificial Intelligence; Association for the Advancement of Artificial Intelligence: Palo Alto, CA, USA, 2018; Volume 32, p. 1. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.