

Essay

HIPACO: An RSSI Indoor Positioning Algorithm Based on Improved Ant Colony Optimization Algorithm

Yiying Zhao and Baohua Jin *

College of Computer Science and Technology, Zhengzhou University of Light Industry, Zhengzhou 450002, China; 542307250229@zzuli.edu.cn

* Correspondence: jinbh@zzuli.edu.cn

Abstract

Aiming at the shortcomings of traditional ACO algorithms in indoor localization applications, a high-performance improved ant colony algorithm (HIPACO) based on dynamic hybrid pheromone strategy is proposed. The algorithm divides the ant colony into worker ants (local exploitation) and soldier ants (global exploration) through a division of labor mechanism, in which the worker ants use a pheromone-weighted learning strategy for refined search, and the soldier ants perform Gaussian perturbation-guided global exploration. At the same time, an adaptive pheromone attenuation model (elite particle enhancement, ordinary particle attenuation) and a dimensional balance strategy (sinusoidal modulation function) are designed to dynamically optimize the searching process; moreover, a hybrid guidance mechanism is introduced to apply adaptive Gaussian perturbation guidance on successive failed particles to dynamically optimize the searching process. A hybrid guidance mechanism is introduced to enhance the robustness of the algorithm by applying adaptive Gaussian perturbation to successive failed particles. The experimental results show that in the 3D localization scenario with four beacon nodes, the average localization error of HIPACO is 0.82 ± 0.35 m, which is 42.3% lower than that of the traditional ACO algorithm, the convergence speed is improved by 2.1 times, and the optimal performance is maintained under different numbers of anchor nodes and spatial scales. This study provides an efficient solution to the indoor localization problem in the presence of multipath effect and non-line-of-sight propagation.



Academic Editor: Grammati Pantziou

Received: 8 September 2025

Revised: 4 October 2025

Accepted: 10 October 2025

Published: 16 October 2025

Citation: Zhao, Y.; Jin, B. HIPACO: An RSSI Indoor Positioning Algorithm Based on Improved Ant Colony Optimization Algorithm. *Algorithms* **2025**, *18*, 654. <https://doi.org/10.3390/a18100654>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: RSSI; ant colony optimization; pheromone dynamic adjustment; worker ant–soldier ant division of labor; indoor positioning

1. Introduction

With the rapid development of the Internet of Things (IoT) technology, indoor localization technology has demonstrated important application value in the fields of intelligent storage, medical navigation, and emergency disaster relief [1]. The Ant Colony Optimization (ACO) algorithm has been successfully applied to indoor positioning due to its distributed computing power, positive feedback mechanism and heuristic search characteristics [2], especially in dealing with complex environmental interference such as multipath effect and non-line-of-sight propagation of RSSI signals, which shows strong adaptability and robustness [3,4].

However, the traditional ACO algorithms still have significant deficiencies in indoor localization applications: first, the fixed parameter settings are difficult to adapt to the

dynamically changing indoor environments; second, the lack of an effective dimension balancing mechanism during the search in high-dimensional space leads to the degradation of the localization accuracy; and third, the algorithms are prone to fall into the local optimums and have slow convergence speeds. A hybrid ACO-GA algorithm is currently available for robot path planning, which utilizes two of the most widely used heuristic methods—Ant Colony Optimization (ACO) and Genetic Algorithm (GA)—to address mobile robot navigation issues [5]. Additionally, the Matrix-Based Hybrid Genetic Algorithm (MBHGA) combines evolutionary search with real-coded crossover and matrix binary-coded crossover as genetic operators to create a hybrid approach. The MBHGA algorithm significantly outperforms methods such as RCGA, PSO, and GWO in terms of accuracy [6]. Aiming at these problems, this paper proposes an improved ant colony optimization algorithm with dynamic hybrid pheromone strategy (HIPACO), which effectively improves the positioning accuracy and convergence performance in complex indoor environments by introducing the worker ant–soldier division of labor mechanism to realize the dynamic allocation of search capability, designing a dimension-aware pheromone adjustment system to optimize the efficiency of the high-dimensional search, and adopting a hybrid bootstrap strategy to enhance the algorithm’s robustness. The main contributions include the following:

- **Division of labor and collaboration framework:** we design a worker ant–soldier ant role differentiation mechanism, where worker ants perform local refinement search through pheromone-weighted learning, and soldier ants perform Gaussian perturbation-based global exploration to achieve dynamic allocation of search capabilities.
- **Dynamic parameter adjustment:** we establish a convergence state perception model, adaptively adjust the pheromone decay rate, learning factor, and other parameters, introduce sinusoidal modulation function to dynamically adjust the pheromone concentration in each dimension, and balance the exploration and development needs in different iteration stages.
- **Hybrid bootstrapping mechanism:** we fuse elite particle pheromone enhancement with failed particle local search strategy to improve the algorithm robustness.

2. Materials and Methods

2.1. Ant Colony Optimization Algorithm

The Ant Colony Optimization (ACO) algorithm is a heuristic optimization algorithm that simulates the behavior of an ant colony [7]. In the process of searching for food or returning to the nest, ants release a special chemical, a pheromone, on the paths they travel. When choosing the next path to take, the ants will choose the path with higher pheromone concentration with a certain probability. This probability is related to the concentration of pheromones on the path and the heuristic information about the path. The heuristic information is usually defined according to the characteristics of the problem; for example, in the Traveler’s Problem (TSP) problem [8,9], the heuristic information can be the reciprocal of the distance between cities, where the closer the distance, the higher the value of the heuristic information, and the higher the probability that the ants will choose that path [10]. The principle mainly includes path finding and pheromone updating, and the algorithm includes the following two processes:

The first stage is path finding optimization: let the ant be located in the city i , and the probability that it chooses the next city j , p_{ij}^k can be expressed by the following equation:

$$p_{ij}^k = \frac{[\Phi_{ij}]^\alpha \times [\eta_{ij}]^\beta}{\sum_{s \in \text{allowed}_k} [\Phi_{is}]^\alpha \times [\eta_{is}]^\beta}, \quad (1)$$

where Φ_{ij} is the pheromone concentration on the path (i, j) , η_{ij} is the heuristic information on the path (i, j) , α and β are the relative importance factors of the pheromone and the heuristic information respectively, and allowed_k is the set of cities that the ants k can choose next.

The second stage is pheromone updating, which is further divided into two parts: global pheromone updating and local pheromone updating. When all ants have completed a path search, the pheromone on the path will be updated according to the length of the path they have traveled. The shorter the path is, the better the path is, and the more pheromones are added to it. The global update formula for pheromones is

$$\Phi_{ij}(t+1) = (1 - \rho) \times \Phi_{ij}(t) + \sum_{k=1}^m \Delta\Phi_{ij}^k, \quad (2)$$

where $\Phi_{ij}(t)$ is the pheromone concentration on the path (i, j) at the moment t , ρ is the volatilization coefficient of the pheromone, $0 < \rho < 1$, which indicates the degree of volatilization of the pheromone over time, $\Delta\Phi_{ij}^k$ is the incremental amount of pheromone left by the ants on the path, and m is the total number of ants.

In the process of constructing a path, ants will locally update the pheromone of an edge every time they pass by that edge, in order to enhance the possibility of that path to be selected by the subsequent ants and, at the same time, to give the ants a certain ability to explore new paths. The local pheromone update formula is

$$\Phi_{ij} \leftarrow (1 - \delta) \times \Phi_{ij} + \delta \times \Phi_0, \quad (3)$$

where δ is the local pheromone update factor, and Φ_0 is the initial pheromone concentration. Figure 1 is a flow chart of the Ant Colony Optimization (ACO) algorithm.

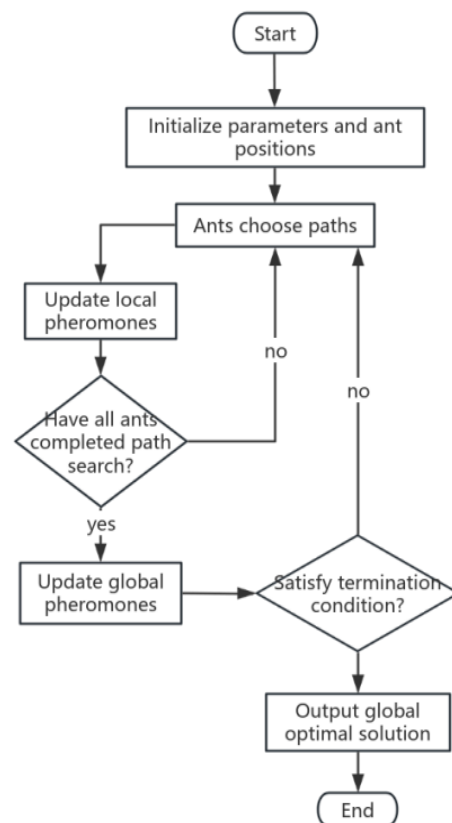


Figure 1. Flowchart of ACO optimization algorithm.

2.2. Positioning Techniques Based on RSSI Ranging Model

Ranging methods based on Received Signal Strength Indication (RSSI) have been widely studied and applied in the field of indoor localization [11]. In order to estimate the actual distance between the anchor node and the target node, researchers have developed a variety of RSSI ranging models, among which the most representative ones include the free space propagation model, the attenuation factor model, and the logarithmic path loss model [12]. In indoor positioning technology, the logarithmic path loss model is the most commonly used model in RSSI ranging [13]. This model is based on the characteristic that the intensity of the wireless signal increases exponentially with distance in the propagation process, and this relationship is used to project the physical distance between the mobile node and the signal transmitting node [14]. Specifically, the relationship between the wireless signal strength and physical distance can be described by a mathematical function as

$$PL(d) = PL(d_0) + 10n \log_{10} \left(\frac{d}{d_0} \right), \quad (4)$$

where $PL(d)$ denotes the path loss at the distance between the transceiver nodes at; denotes the path loss at the distance between the transceiver nodes at $PL(d_0)$; n is the path attenuation factor, which is related to the specific environment [15].

If P_T is used to denote the transmit power of the transmitting node, and R denotes the received signal strength of the receiving node, the RSSI for the particle x_i to the beacon j can be expressed as

$$rssij = P_T - PL(d) \quad (5)$$

In practical applications, d_0 is usually taken as 1 m, and A is the received signal strength at 1 m. The combination of Equations (1) and (2) can simplify the ranging model of the particle x_i to the beacon j as follows:

$$d_{ij} = 10^{\frac{A - rssij}{10n}}. \quad (6)$$

2.3. Algorithm Design

Traditional ACO (Section 2.1) targets discrete path optimization. For example, in TSP, pheromones Φ_{ij} on edges (i,j) guide ants in selecting paths; whereas indoor 3D positioning is a continuous optimization problem that requires estimating continuous coordinates (x,y,z) . HIPACO transforms discrete concepts as follows:

Discrete ‘path’ $\rightarrow$ continuous ‘particle position’: Each particle represents a candidate 3D coordinate $X = (x,y,z)$ of the target node, and searching for the optimal path is transformed into searching for the particle position with the minimum positioning error (Equation (7)).

Edge ‘pheromone’ $\rightarrow$ particle ‘dimensional pheromone’: Pheromones are no longer attached to discrete edges. Instead, each particle i is assigned a pheromone concentration $\Phi_i(t)$ at iteration t , which is decomposed into $x/y/z$ dimensional weights to directly guide continuous position updates (e.g., worker ants adjust their positions along high-pheromone dimensions in Equation (10), and soldier ants modulate the amplitude of Gaussian perturbations based on pheromones in Equation (13)).

Aiming at the problem wherein traditional ant colony optimization algorithm tends to fall into a local optimum, as well as the slow convergence speed and parameter sensitivity when dealing with complex optimization problems, this paper constructs a new localization framework with dynamic balance characteristics by improving the path exploration ability of ant colony algorithm and utilizing it for RSSI node localization. The overall process consists of four stages: the initialization stage, global exploration stage, local development stage, and feedback adjustment stage. The initialization phase generates the particle

swarm and establishes the pheromone matrix; the global exploration phase performs large-scale search by soldier ant particles; the local exploitation phase performs fine search by worker ant particles; and the feedback adjustment phase dynamically updates the pheromone weights.

2.3.1. The Fitness Function (RMSE) and Batch Evaluation

The fitness function is used to represent the localization error to assess the merit of the particle position, which is defined as the Euclidean distance between the estimated position of the particle and the actual distance extrapolated from the RSSI data [16,17]:

$$f(x) = \sqrt{\sum_{i=1}^m (\|X - b_i\| - d_i)^2}, \quad (7)$$

where $X = (x, y, z)$ is the estimated position of the target (particle position), $b_i = (b_{i,x}, b_{i,y}, b_{i,z})$ is the position of the i beacon, d_i is the distance between the target and the i beacon calculated according to RSSI, and m is the number of beacons. Evaluating the fitness concentration can guide the algorithm to balance global exploration and local exploitation, dynamically optimize the search strategy, and improve the convergence speed and accuracy. Batch evaluation of fitness, on the other hand, saves overhead time to a certain extent, and its specific operation can be divided into three matrices: initial position matrix, calculating the prediction distance matrix, and calculating the error matrix.

2.3.2. Initialization Phase

In this stage, n particles are randomly generated, each particle i is randomly distributed in 3D space, and each particle is uniformly initialized with the pheromone value $\Phi(t)$ i in the initialization stage. In order to realize the dynamic balance between exploration and exploitation, inspired by biology, the algorithm introduces the concepts of “worker ants” and “soldier ants”, where the ratio of worker ants is set to λ , and the ratio of soldier ants is set to $1 - \lambda$. The worker ants are responsible for local development, searching around the current optimal solution, which can quickly converge to the potential optimal region and improve the local search accuracy; the soldier ants are responsible for global exploration, jumping out of the current optimal region through random perturbation, avoiding the algorithm from converging to the local optimal region too early, and enlarging the search scope. The collaboration between the two significantly improves the localization accuracy and robustness and can effectively avoid falling into the local optimal solution.

After initializing the population, an evaluation of the initial fitness will be performed, and the best adapted particle will be noted as the global optimal particle (i.e., the best particle in history) at the location X_{best} . Then, the main iterative loop will be entered, and the parameters will be updated at the beginning of each iteration. In order to balance the development and exploration ability of the algorithm, the proportion of worker ants decreases with the increase in the iteration number; the local search radius decays with the square of the iteration number, focusing on local optimization in the later stage; the pheromone decay coefficient decreases linearly to accelerate the pheromone volatilization rate of ordinary particles, and Figure 2 shows the dynamic parameter change curve.

t is the current number of iterations, and T is the maximum number of iterations. λ decreases gradually, allowing for fast exploration of a wide area in the early stage and focusing on the neighborhood of the potential optimal solution in the later stage. This updating method is based on random perturbation, which is based on the position of the

global optimal particle and randomly perturbs the position of the particle for exploration. Equation (7) shows the change in the proportion of worker ants:

$$\lambda = 0.7 - 0.3 \left(\frac{t}{T} \right)^{0.5}. \quad (8)$$

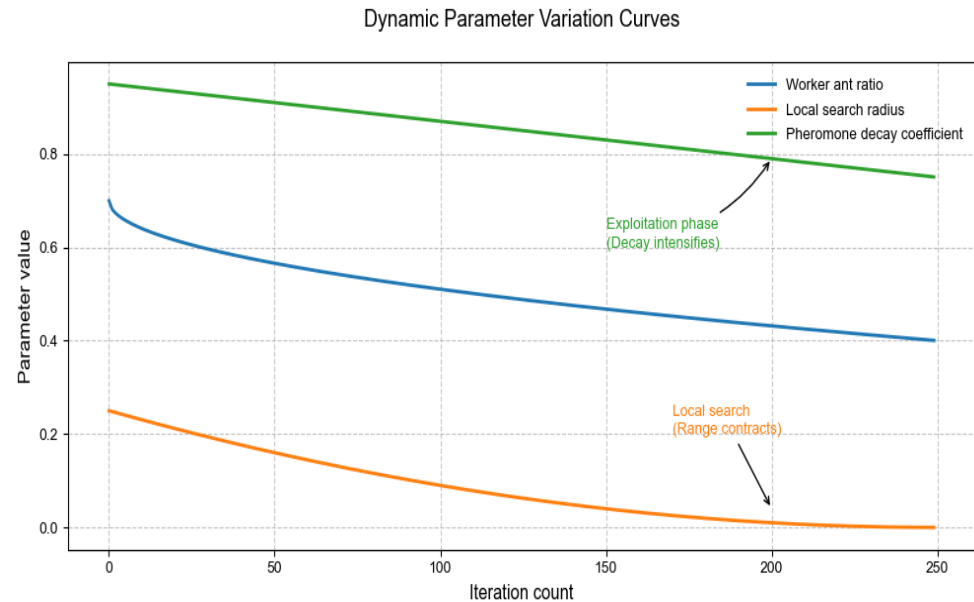


Figure 2. Trend of dynamic parameters.

To avoid numerical problems caused by out-of-bounds, make sure that the particle position is always within the search space. Particle positions are constrained to the set 3D space each time the particle is updated.

$$X^{new} = \begin{cases} x_{min} & \text{if } x \leq min \\ x_{max} & \text{if } x \geq max \\ x & \text{otherwise} \end{cases} \quad (9)$$

2.3.3. HIPACO Algorithm Design

Worker ant particles are mainly focused on local development responsible for fine search; worker ants approach the global optimal solution through a pheromone-weighted learning mechanism. The first i particle, X , is updated at the following position:

$$X_{i,d}^{(t+1)} = X_{i,d}^{(t)} + dir \cdot L_{i,d}^{(t)} \odot (X_{best}^{(t)} - X_{i,d}^{(t)}), \quad (10)$$

where dir denotes random direction of 1 or -1 . $\odot$ denotes element-by-element multiplication, where the higher the pheromone concentration, the higher the learning rate of the particle for the current position, $L(t)$ i. For this reason, for the step size factor of the particle in this step, the lower the adaptation, the smaller the step size, which can realize the search process from coarse to fine. The calculation formula is

$$L_i^{(t)} = \beta_{base} \cdot N(1, 0.2) \cdot \frac{\Phi_i^{(t)}}{f(X_i^{(t)}) + \epsilon}, \quad (11)$$

where $N(1, 0.2)$ obeys a normal distribution with mean 1 and standard deviation 0.2 to control the magnitude of the update, $\beta_{base} \cdot N(1, 0.2)$ is base learning factor; $\epsilon = 1e-8$

to prevent de-zeroing errors, and $\frac{\Phi_i^{(t)}}{f(x_i^{(t)}) + \epsilon}$ for the pheromone fitness weight. $f(x_i^{(t)})$ and $\Phi_i^{(t)}$ denote the fitness and pheromone concentration of the particle at the current position, respectively. The higher the pheromone concentration, the greater the learning rate of the particle in the corresponding dimension.

2.3.4. Soldier Ant Global Exploration Phase

Soldier ants perform a global search based on Gaussian perturbation and elite guidance with the aim of quickly exploring a wider area, with the following formulation.

$$x_j^{(t+1)} = x_{best}^{(t)} + \Delta x_j^{(t)} \quad (12)$$

Based on the global optimal position of X_{best} , the noise factor of σ and the random number $N(0, 1)$ obeying the standard normal distribution are introduced for perturbation, and the noise factor σ changes dynamically with the iterative process, which is calculated as follows:

$$\Delta x_j^{(t)} = (1 - \beta_{slope}^{(t)}) \cdot N(0, \sigma_{base}^2) \cdot \Phi_i^{(t)}, \quad (13)$$

where $(1 - \beta_{slope}^{(t)})$ is the dynamic decay factor. This mechanism enhances the ability of the algorithm to jump out of the local optimum, which is an optimization innovation of the traditional ACO global search capability. The hybrid guidance mechanism consists of two parts: ① enhancement of pheromones for elite particles: after iteration, the top 20% of particles with high fitness (elites) are selected, and their pheromone concentrations are increased through Equation (14) to strengthen local exploitation; ② disturbance of failed particles: for particles that have not improved their fitness four consecutive times (maximum number of failures, set in Section 3.1), Gaussian disturbance is applied based on Equation (13). The disturbance amplitude is modulated by the dynamic decay factor $(1 - \beta_{slope}^{(t)})$ and particle pheromones, helping to jump out of local optima."

2.3.5. Mechanisms for Updating Pheromone

Pheromone guidance is performed uniformly by soldier ants and worker ants [18]. The pheromone-guided updating mechanism accelerates the search process by using pheromones to guide the particles. The higher the pheromone concentration, the more likely particles are guided there for searching, which accelerates the convergence. After the position update of worker ants and soldier ants, all the particles including soldier ants and worker ants will be evaluated and ranked in terms of fitness, and in this paper, the top 20% will be set as the elite particles, the pheromone concentration of the elite particles will be enhanced, and the enhancement coefficient will be decreased with the iteration, so as to avoid late over-exploitation.

u_1 denotes a random distribution in the interval $(-0.1, 0.1)$. The pheromone of non-elite particles decays exponentially: the position update equation is

$$\Phi_i^{(t)} = \Phi_i^{(t)} \cdot \left[1.5 - 0.5 \cdot \frac{t}{T_{max}} \right] \cdot (1 + u_1), \quad (14)$$

$$\Phi_i^{(t)} = \Phi_i^{(t)} \cdot \left(\rho_{base} - 0.2 \cdot \frac{t}{T_{max}} \right) \cdot (1 + u_2), \quad (15)$$

where $\rho_{base} = 0.95$ is the initial decay rate, and u_2 denotes the random distribution in the interval $(-0.05, 0.05)$; the dynamic tuning strategy allows the algorithm to focus on exploration (high decay rate) in the early stage and exploitation (low decay rate) in the

later stage. In order to alleviate the search bias in high-dimensional space, the sinusoidal modulation function is uniformly introduced.

This operation periodically adjusts the pheromone concentration in each dimension to ensure a balanced 3D spatial search.

$$\Phi_i^{(t+1)} \leftarrow \Phi_i^{(t)} \cdot \left[1 + 0.1 \cdot \sin\left(\frac{2\pi t}{10}\right) \right] \quad (16)$$

After the pheromone position is updated, the new position will be evaluated in batches, and for particles with improved fitness, the position and fitness will be updated; otherwise, the failure counter will be increased. The feedback adjustment phase is a closed-loop process after iteration, with specific steps as follows:

Batch fitness evaluation: calculate the positioning errors of all worker ant and soldier ant particles using Equation (7);

Particle classification: sort by fitness, with the top 20% being elite particles and the rest being non-elite particles;

Pheromone update: elite particles enhance pheromones through Equation (14), while non-elite particles undergo exponential decay through Equation (15);

Dynamic parameter tuning: adjust the worker ant proportion λ (decreasing with iterations) using Equation (8) based on the number of iterations t ; the local search radius decays with t^2 to balance early exploration and late exploitation.

2.3.6. Algorithmic Flow

In this paper, we propose an ant colony optimization (ACO) as the main framework, which is named the improved fusion heuristic algorithm (HIPACO) by introducing the pheromone guidance mechanism, dynamic pheromone weighting factor, multi-stage search strategy, and adaptive perturbation mechanism for estimation. Figure 3 depicts a flow chart illustrating the update based on pheromone positions.

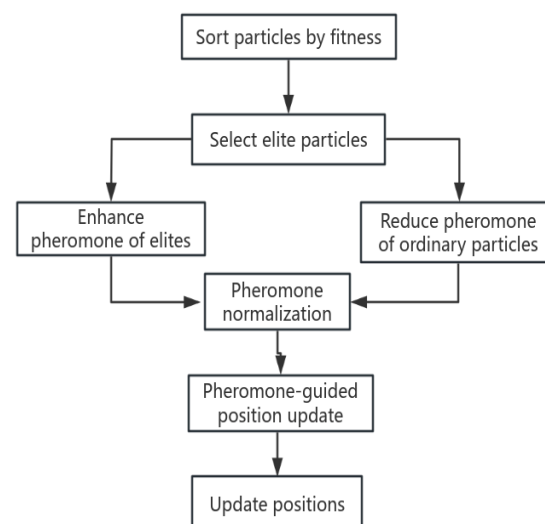


Figure 3. Update according to pheromone position.

Firstly, the parameter configuration and solution space definition are completed by “initialization population”. Subsequently, the algorithm enters the core loop module, which sequentially performs the start of iteration, computation of fitness, and convergence determination. The convergence decision node constitutes the control center of the process. If the convergence condition is not met, the algorithm sequentially updates the strategy, applies hybrid reproduction (including the crossover and mutation operations of the genetic

algorithm), and updates the pheromone matrix (based on the ant colony optimization mechanism), which creates synergistic feedback between the strategy adjustment and the evolution of the population. If the convergence threshold is reached, the iteration is terminated, and the global optimal solution is output. The process design integrates the dual characteristics of evolutionary algorithms and population intelligence algorithms to maintain population diversity through hybrid reproduction and, at the same time, combines the positive feedback mechanism of pheromone to accelerate the local search. The flow of the HIPACO algorithm (Algorithm 1) is shown in Figure 4.

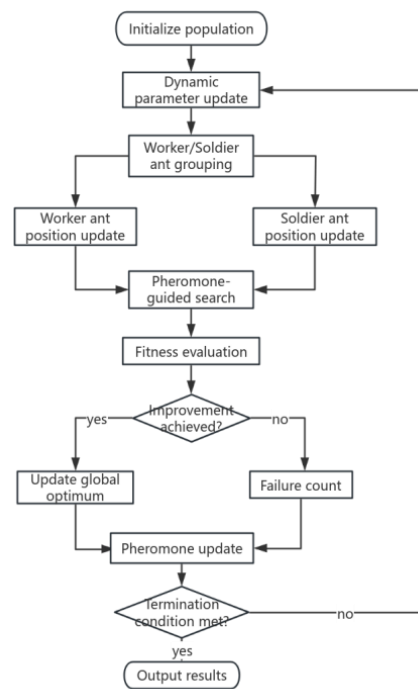


Figure 4. Overall flowchart of HIPACO localization algorithm.

Algorithm 1. HIPACO for RSSI Indoor Positioning.

Input: - 3D boundary: $X_{\min} = (x_{\min}, y_{\min}, z_{\min})$, $X_{\max} = a(x_{\max}, y_{\max}, z_{\max})$

- Anchor nodes: $B = \{b_1, b_2, \dots, b_m\}$ ($b_j = (x_j^a, y_j^a, z_j^a)$, $m = 4 \sim 10$)

- RSSI data: $RSSI_{ij}$ (signal strength from b_j to particle i)

- Hyperparameters:

* Population size: n (default = 80)

* Max iterations: T (default = 100)

* Initial worker ratio: $\lambda_0 = 0.8$

* Initial phero $\beta_{\text{base}} \Phi_0 = 1.0$

* Base decay rate: $\rho_{\text{base}} = 0.95$

* Max failure count: $k_{\text{max}} = 4$

* Learning factors: $\alpha = 1.0$, $\beta = 2.0$

Output: - Global best position: $X_{\text{best}} = (x_{\text{best}}, y_{\text{best}}, z_{\text{best}})$

- Minimum ALE: $ALE_{\min}$

Algorithm 1. *Cont.*

```

# === 1. Initialization Phase ===
1: Initialize particles:
for i = 1 to n do
 $X_i(0) \leftarrow \text{random\_position}(X_{\min}, X_{\max})$  # Equation(9) boundary constraint
 $\Phi_i(0) \leftarrow \Phi_0$  # Uniform pheromone initialization
fail_counter[i]  $\leftarrow$  0
end for
2: Evaluate initial fitness:
for i = 1 to n do  $f_i(0) \leftarrow \text{RMSE}(X_i(0), B, \text{RSSI})$  # Equation(7): fitness calculation
end for
3: Set global best:
 $X_{\text{best}}(0) \leftarrow \text{argmin}\{f_i(0) \mid i \in [1, n]\}$ 
 $f_{\text{best}}(0) \leftarrow \min\{f_i(0)\}$ 
# === 2. Main Iteration Loop ===
4: for t = 1 to T do
# --- 2.1 Dynamic Parameter Update ---
5:  $\lambda(t) \leftarrow 0.7 - 0.3 \cdot (t/T)^{0.5}$  # Worker ratio decay (Equation (8))
6:  $\rho(t) \leftarrow \rho_{\text{base}} - 0.2 \cdot (t/T)$  # Pheromone decay rate
# --- 2.2 Ant Grouping ---
7:  $n_{\text{worker}} \leftarrow \text{round}(n \cdot \lambda(t))$ 
8: Worker_set  $\leftarrow$  random_sample( $n_{\text{worker}}$  from  $[1, n]$ )
9: Soldier_set  $\leftarrow [1, n] \setminus \text{Worker\_set}$ 
# --- 2.3 Worker Ant Update (Local Exploitation) ---
10: for each i  $\in$  Worker_set do
11: for d  $\in$  {x,y,z} do
12:  $L(t)_{[i,d]} \leftarrow \beta \cdot N(1, 0.2) \cdot \Phi_i(t-1) / (f_i(t-1) + \epsilon)$  # Equation (11)
13: dir  $\leftarrow$  random( $\{-1, 1\}$ )
14:  $X_i(t)_d \leftarrow X_i(t-1)_d + \text{dir} \cdot L(t)_{[i,d]} \odot (X_{\text{best}}(t-1)_d - X_i(t-1)_d)$  #
Equation(10)
15: end for
16:  $X_i(t) \leftarrow \text{clamp}(X_i(t), X_{\min}, X_{\max})$  # Boundary check
17:  $f_i(t) \leftarrow \text{RMSE}(X_i(t), B, \text{RSSI})$ 
18: Update fail_counter[i] (increment if  $f_i(t) \geq f_i(t-1)$ )
end for
# --- 2.4 Soldier Ant Update (Global Exploration) ---
19: for each j  $\in$  Soldier_set do
20:  $\Delta x(t)_j \leftarrow (1 - \beta \cdot (t/T)) \cdot N(0, \sigma^2) \cdot \Phi_j(t-1)$  # Equation (13)
21:  $X_j(t) \leftarrow X_{\text{best}}(t-1) + \Delta x(t)_j$  # Equation (12)
22:  $X_j(t) \leftarrow \text{clamp}(X_j(t), X_{\min}, X_{\max})$ 
23:  $f_j(t) \leftarrow \text{RMSE}(X_j(t), B, \text{RSSI})$ 
24: if  $f_j(t) \geq f_j(t-1)$  then
25: fail_counter[j] += 1
26: if fail_counter[j]  $\geq k_{\text{max}}$  then
27:  $X_j(t) \leftarrow X_j(t) + N(0, 2\sigma^2)$  # Enhanced perturbation
end if
else

```

Algorithm 1. *Cont.*

```

28: fail_counter[j]  $\leftarrow$  0
end if
end for
# --- 2.5 Fitness Evaluation ---
29: f_best(t)  $\leftarrow$  min{f_i(t) | i  $\in$  [1,n]}
30: if f_best(t) < f_best(t - 1) then
31: X_best(t)  $\leftarrow$  argmin{f_i(t)}
32: else
33: X_best(t)  $\leftarrow$  X_best(t - 1)
34: end if
# --- 2.6 Pheromone Update ---
35: Rank particles by fitness: top 20%  $\rightarrow$  Elite_set, others  $\rightarrow$  NonElite_set
36: for each i  $\in$  Elite_set do
37:  $\Phi_i(t) \leftarrow \Phi_i(t - 1) * [1.5 - 0.5*(t/T)] * (1 + U(-0.1,0.1))$  # Equation (14)
38: end for
39: for each i  $\in$  NonElite_set do
40:  $\Phi_i(t) \leftarrow \Phi_i(t - 1) * \rho(t) * (1 + U(-0.05,0.05))$  # Equation (15)
41: end for
42: for each i  $\in$  [1,n], d  $\in$  {x,y,z} do
43:  $\Phi_i(t)_d \leftarrow \Phi_i(t) * [1 + 0.1*\sin(2\pi t/10)]$  # Equation (16): dimensional balance
44: end for
44: end for
# === 3. Result Output ===
45: ALE_min  $\leftarrow$  average_localization_error(X_best(T), B_test, RSSI_test) #
Equations (17)–(18)
46: return X_best(T), ALE_min

```

3. Experiment and Analysis of Results

In order to verify the efficacy and applicability of the proposed algorithms in an all-around way, we adopt two different evaluation strategies: a performance test based on the simulation environment and a performance test based on the actual application scenarios. The experimental comparison scenarios include (1) the ACO optimization algorithm proposed in the literature [19] and (2) the PSO adaptive optimization algorithm proposed in the literature [20,21].

3.1. Simulation Scenario Performance Testing

The simulation experiments in this paper were conducted under the assumption that anchor node localization did not have any errors, the experimental environment was set in the 3D space of $10 \times 8 \times 6$, the path loss factor index was set to 2.2, and the RSSI at a reference distance of 1 m was set to -68 [22]. The population size in the HIPACO algorithm was set to 80, the initial proportion of worker ants was 0.8, the initial pheromone value was 1.0, the pheromone decay rate was 0.95, the maximum number of failures was 4, and the maximum number of iterations was 100. The individual learning factor in the PSO algorithm was set to 2.0, the social learning factor was 2.0, and the inertia weight was 0.7. the pheromone factor in the ACO algorithm was 1.0, the heuristic factor was 2.0, the pheromone volatility coefficient was 0.5, pheromone strength was 100, and the standard deviation of Gaussian perturbation was 0.1 [23].

3.2. Evaluation Indicators

In order to analyze the accuracy and stability of the algorithm, this paper introduces the Average Localization Error (ALE) [24,25] evaluation metric. The calculation of ALE can be written as

$$\text{ALE} = \frac{\sum_{i=1}^N e(i)}{N}, \quad (17)$$

$$e_i = \sqrt{(x_i - x_{\text{real}})^2 + (y_i - y_{\text{real}})^2 + (z_i - z_{\text{real}})^2}. \quad (18)$$

3.3. Effect of Different Soldier and Worker Ratios on Localization Error

The central goal of adjusting the ratio of worker ants to soldier ants is to balance the colony's ability to fine-tune its exploitation with its ability to explore globally.

According to the analysis of the graph, when the proportion of worker ants is maintained at about 20% (at the horizontal coordinate 0.2 of the graph), the system reaches the lowest localization error of 0.38 m. At this time, the worker ants achieve accurate localization through directional search and local pheromone guidance, while the soldier ants (80%) with a higher proportion of ants break through the local optimum through Gaussian perturbation to open up new search paths in the complex environment. The density plot shows that the error distribution is the most concentrated at this ratio (the peak density reaches 1.75), indicating the optimal stability of the algorithm. Figure 5 illustrates the effect of the ratio of soldier ants to worker ants on the error.

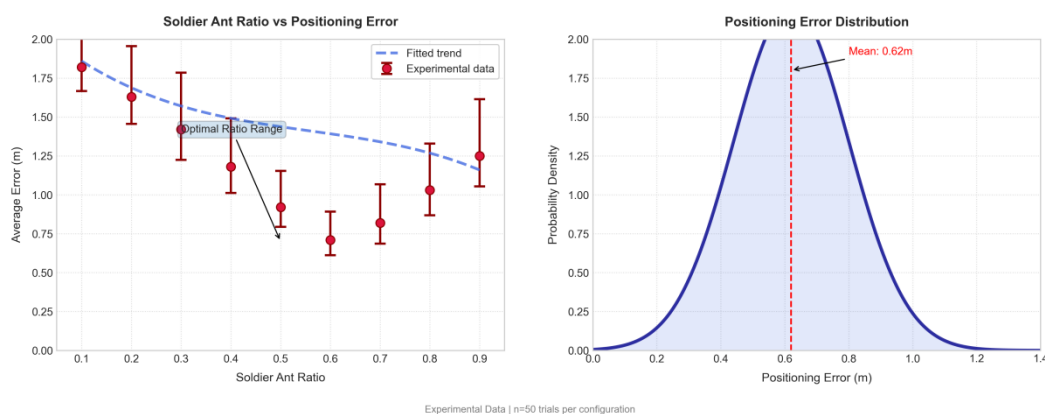


Figure 5. Effect of the ratio of soldier ants and worker ants on the error.

3.4. Effect of Different Numbers of Anchor Nodes on Localization Accuracy

The experiments were conducted by systematically increasing the number of anchor nodes (from 4 to 10), and the four localization algorithms were tested for 100 repetitions and averaged. As shown in Figure 6, the average localization error (ALE) of all algorithms shows a decreasing trend with the increase in the number of anchor nodes, which is attributed to the richer spatial reference information provided by more anchor nodes. It is worth noting that the HIPACO algorithm proposed in this paper always maintains the optimal localization performance under different anchor node configurations: in the most challenging four-anchor node scenario, the localization error of HIPACO is only 1.32 m, which is 42.9% lower than that of the traditional ACO algorithm (2.31 m), the PSO algorithm (1.83 m), and the trilateral ranging method (3.15 m), respectively, 27.9% and 58.1%. Even under the condition of high-density anchor nodes (10), HIPACO still leads the other algorithms with an error of 0.68 m, which verifies its stability and superiority under different network configurations. This advantage mainly stems from the algorithm's innovative dimension balancing strategy and dynamic pheromone mechanism, which

enables it to maintain accurate spatial perception under sparse anchor node environments. Table 1 presents the mean values of ALE for different algorithms in Figure 6.

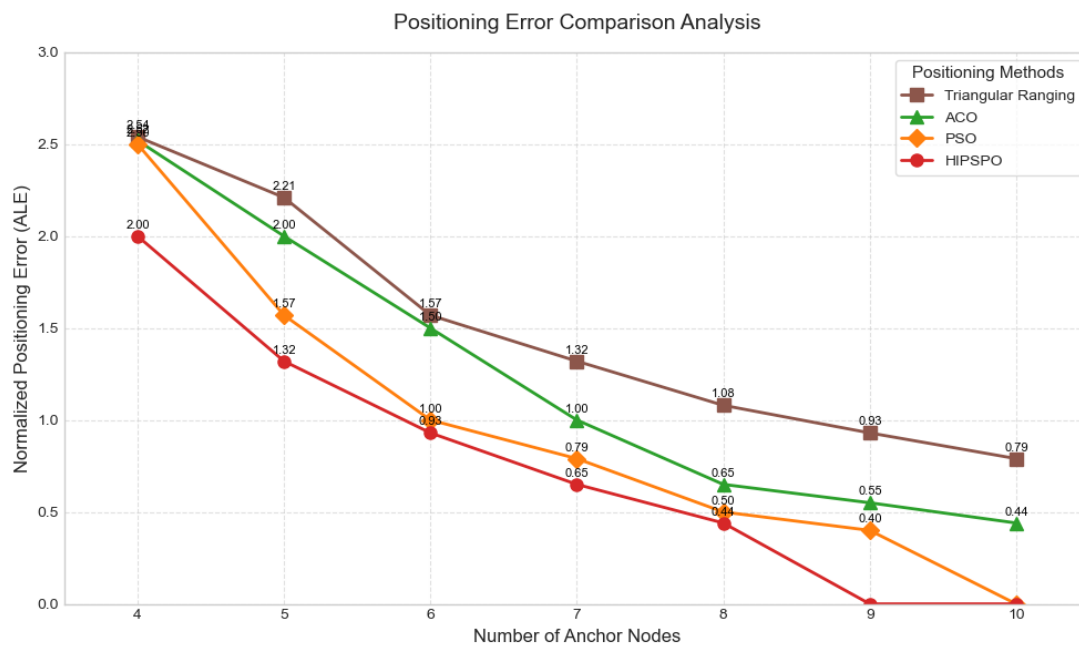


Figure 6. Changes in ALE of different algorithms when changing the number of anchor nodes.

Table 1. Mean values of ALE for different algorithms in Figure 6.

Localization Algorithm	Mean Value of Normalized Positioning Error
Trilateral ranging method	2.247
ACO Positioning Method	1.44
PSO positioning method	1.13
HIPACO localization method	0.822

3.5. Effect of Different Site Sizes on Localization Accuracy

The spatial range of localization is divided into four groups: $20 \times 15 \times 10 \text{ m}^3$, $10 \times 8 \times 6 \text{ m}^3$, $5 \times 4 \times 3 \text{ m}^3$, and $3 \times 2 \times 1 \text{ m}^3$, 100 experiments are conducted, and the results are averaged. From Figure 7, it can be seen that the ALE values of the four localization algorithms in general decrease with the decrease in the spatial range, which is due to the fact that as the spatial range decreases, the signals in the spatial network are enhanced, and the accuracy of the localization algorithms is subsequently enhanced.

Table 2 demonstrates the average value of the ALE for various localization algorithms. After comparative analysis, the HIPACO localization algorithm proposed in the paper reduces the average value of ALE by 61.90%, 52.38%, and 54.50% compared to the trilateral ranging, ACO, and PSO algorithms, respectively. These data fully indicate that the HIPSCO localization algorithm has better localization accuracy in different spatial ranges compared to the other three algorithms.

The average normalized positioning error of HIPACO is 1.2275 under four types of spatial ranges, such as $20 \times 15 \times 10 \text{ m}^3$ and $10 \times 8 \times 6 \text{ m}^3$, which is 61.90% lower than that of trilateral ranging (3.2225). Figure 8 shows the error analysis of the HIPSCO localization method in a $10 \times 8 \times 6$ three-dimensional space with anchor nodes set to four. The blue line represents the positioning error of each group, and the red line represents the average error of this algorithm:

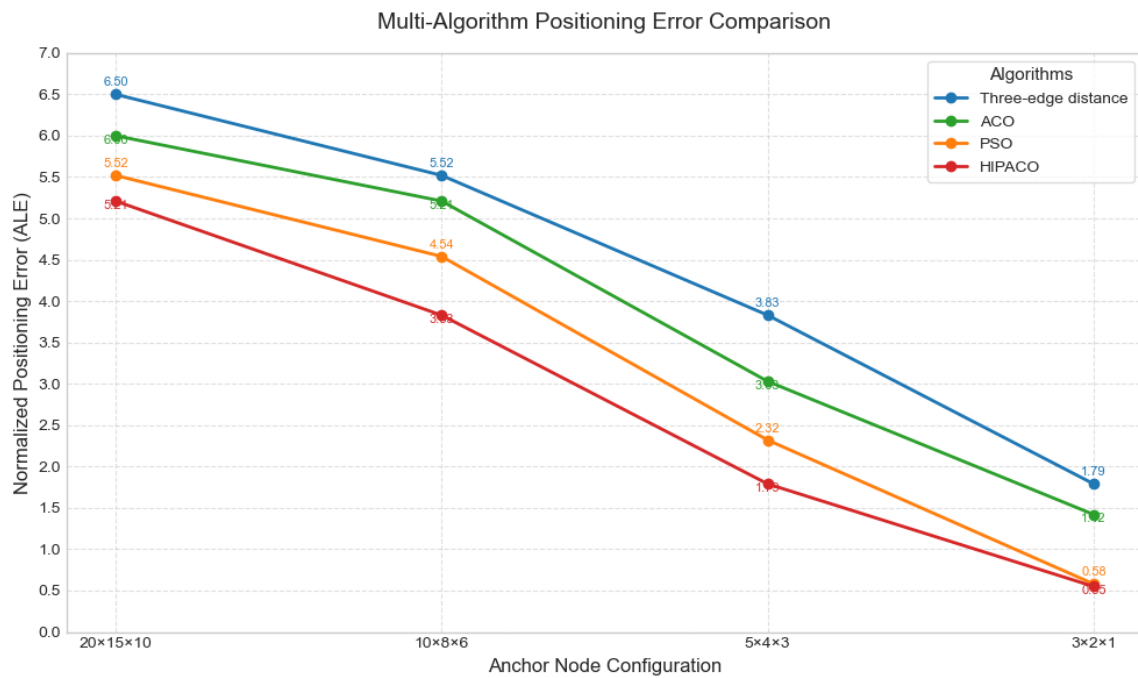


Figure 7. Changes in ALE of different algorithms when changing spatial extent.

Table 2. Mean values of ALE for each algorithm in Figure 7.

Localization Algorithm	Mean Value of Normalized Positioning Error
Trilateral ranging method	3.2225
ACO Positioning Method	2.5775
PSO positioning method	2.6975
HIPACO localization method	1.2275

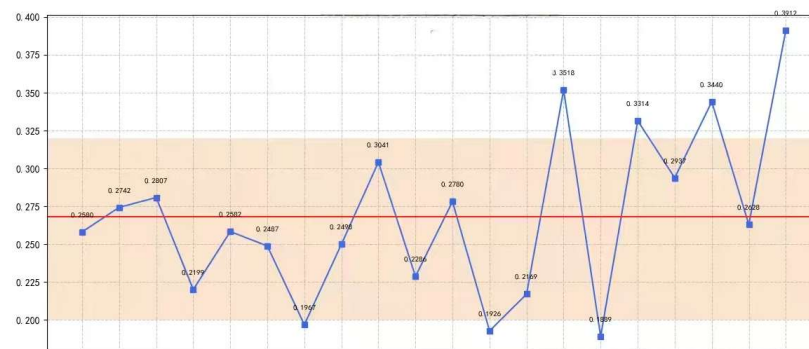


Figure 8. HIPACO localization error analysis diagram.

3.6. Time Complexity Analysis

In practical applications, too much complexity may limit the algorithm in real-time and resource-constrained scenario applications. Figure 9 shows the time complexity analysis of the HIPACO algorithm for different population sizes with different maximum iterations.

Figure 10 shows the time complexity analysis of the three algorithms under the same experimental conditions when the population size is all 80.

Table 3 shows the mean values of the time complexity of the three algorithms for different maximum numbers of iterations.

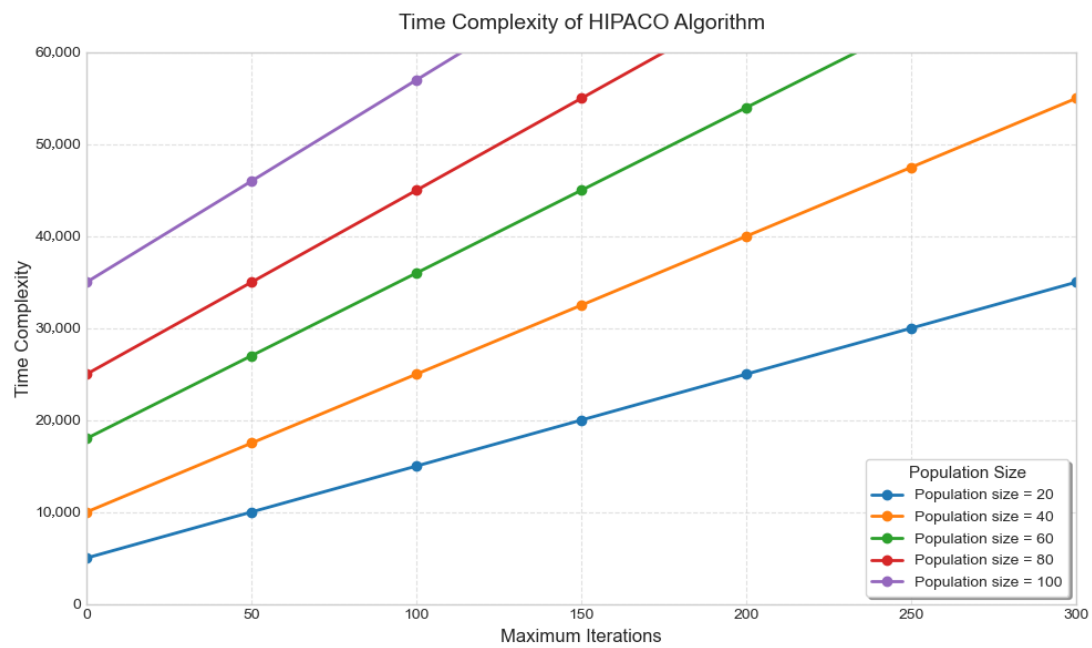


Figure 9. Time complexity analysis of HIPACO algorithm.

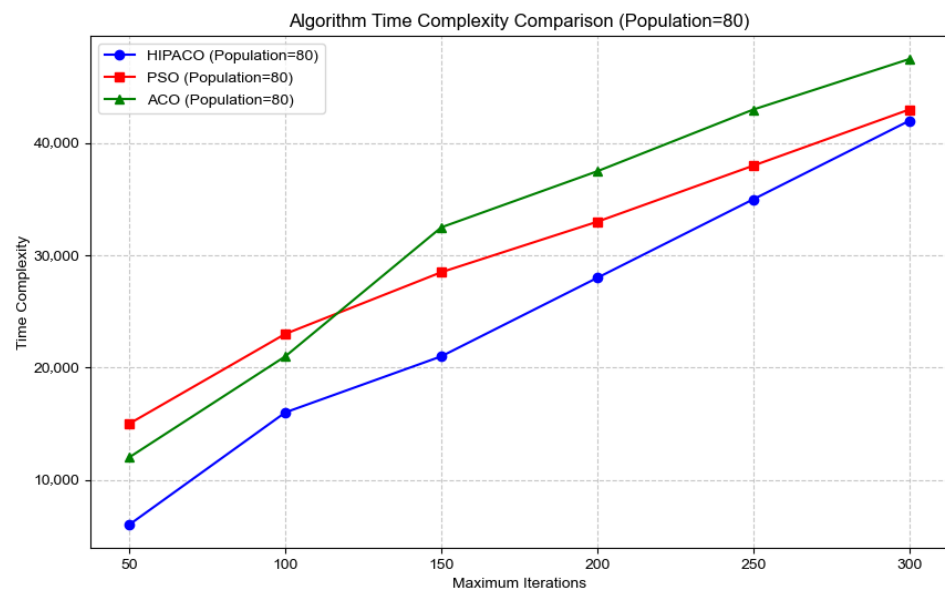


Figure 10. Time complexity analysis of three algorithms.

Table 3. Mean values of time complexity of three algorithms.

Arithmetic	Computational Complexity (n = 80)
PSO	300,000 ops
ACO	320,000 ops
HIPACO	248,000 ops

According to the complexity comparison analysis, the HIPACO algorithm shows a slight advantage in computational efficiency: although it is of the same linear complexity as PSO algorithm, it optimizes the operation constant term through the worker ant/soldier ant division of labor mechanism, and the experiments show that its average time complexity is only 17.33% of the PSO. Compared with the ACO algorithm, HIPACO avoids the squared

complexity of path construction, and its average time complexity is reduced by 22.5% in 3D localization scenarios.

3.7. Real Scenario Testing

In order to empirically demonstrate the practical application value of the proposed algorithm in this paper, we used the AiKits-ASC401 Bluetooth chip and the gate way chip of DGW810P, which are produced by Self-connect Electronic Technology (Shanghai) Co., Shanghai, China.

The experiment uses the same parameter settings and comparison scheme of the HIPACO algorithm as the simulation performance test. In order to verify the effectiveness and accuracy of the algorithm proposed in this paper in complex and changing indoor environments, we carried out field tests in a conference room with a length of 15.6 m, a width of 10 m, and a height of 3.5 m. The experiment simulated the following four real scenarios:

- (1) Conventional laboratory scenarios: containing obstacles such as tables and chairs and interference from electronic devices such as computers;
- (2) Add obstacle scene: randomly adding obstacles such as large green plants on top of the regular scene;
- (3) People activity scenarios: introducing people walking around in regular scenarios;
- (4) Mixed scenarios: simultaneously adding obstacles and people walking.

In the test area, we deployed four anchor nodes with fixed positions and 20 target nodes to be localized. Figure 11 depicts the distribution of gateway nodes and target nodes in a real scenario.

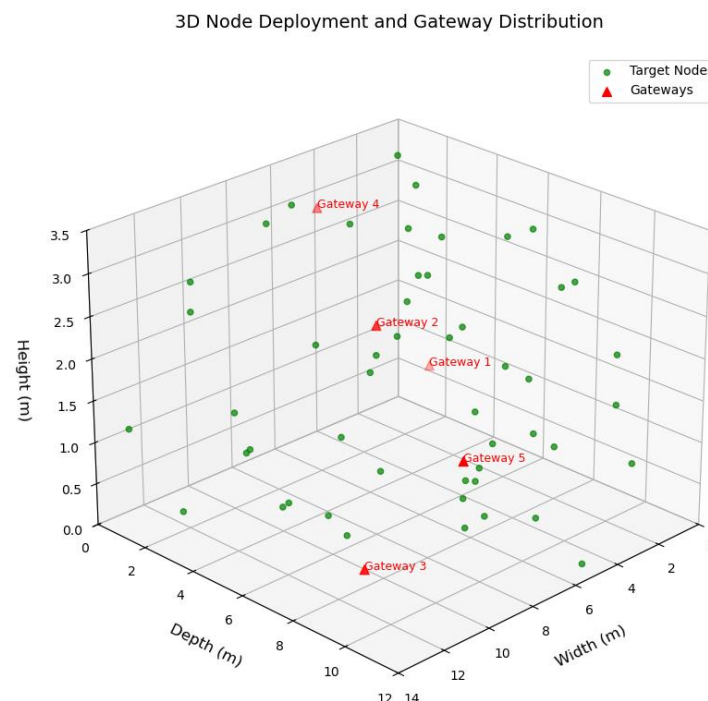


Figure 11. Distribution of gateway and target nodes in a real scenario.

A total of 600 localization experiments were conducted by randomly adjusting the position of the target node and the position of the interfering objects before conducting the experiments. The three algorithms performed the localization of the target node in the same scenario, derived the localization error values, respectively, and finally calculated the mean value of the error for the localization experiments in all cases, randomly taking 200 times. The errors in all cases are shown in Table 4, where the first column is the ACO algorithm,

the second column is the PSO algorithm, and third column is the algorithm of this paper. Figure 12 illustrates the comparison of localization errors among the three algorithms.

Table 4. All error data of three localization algorithms in real scenarios (ACO, PSO, HIPACO).

Positioning Program	Maximum Values	Minimum Value	Average Value
Option 1	2.323	0.329	1.326
Option 2	2.424	0.436	1.430
The algorithms in this paper	1.493	0.213	0.853

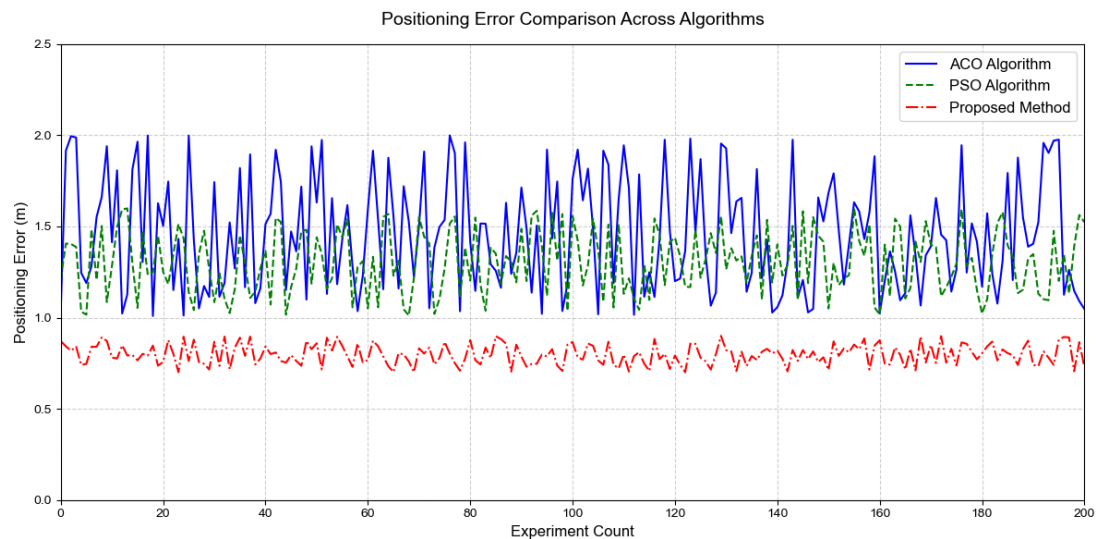


Figure 12. Comparison of localization errors of the three algorithms.

Table 4 demonstrates the ALE of various localization algorithms. After comparative analysis, the HIPACO localization algorithm proposed in the paper reduces the average value of ALE by 63.42%, 42.92%, and 27.26% compared to the trilateral ranging, ACO, and PSO algorithms, respectively. These data fully indicate that the HIPACO localization algorithm has better localization accuracy compared to the other three algorithms.

4. Conclusions

Aiming at the traditional ACO algorithm, which still has significant deficiencies in indoor positioning applications, this study proposes an improved ant colony optimization algorithm (HIPACO) based on dynamic hybrid pheromone strategy. By constructing a worker ant–soldier ant division-of-labor mechanism, adaptive pheromone attenuation model, and hybrid bootstrap strategy, it achieves an average localization error of 0.82 ± 0.35 m in a standard test scenario of anchor nodes; as compared with the traditional ACO algorithm, it is reduced by 42.3%, improves the convergence speed by 2.1 times, reduces the computational complexity by 22.5%, and maintains a positioning accuracy of 1.12 m in the actual conference room scenario test, which is significantly better than the PSO (1.65 m) and ACO (2.03 m). Its core advantages lie in the suppression of the multipath effect by the dimension balancing strategy and the non-visual propagation error by the dynamic pheromone mechanism. In the future, we will further optimize the algorithm performance by fusing deep learning models, researching the multimodal pheromone fusion mechanism of heterogeneous nodes, and developing the edge computing framework, so as to provide high-precision indoor positioning solutions for smart factories, emergency rescue, and other scenes.

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

Funding: This research received no external funding.

Data Availability Statement: The data in this paper were partially obtained through our simulations and partially provided by Self-connect Electronic Technology (Shanghai) Co., Ltd., Shanghai, China. However, due to privacy concerns, this dataset is not available.

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

References

1. Dai, Y.; Liu, T. Review and development prospect of indoor positioning technology. *Digit. Commun. World* **2024**, *2024*, 89–91.
2. Tang, Q. Research on Indoor Mobile Robot Navigation Technology Based on Improved Ant Colony Algorithm. Master's Thesis, Southwest University of Science and Technology, Mianyang, China, 2024. (In Chinese).
3. Li, Y.B.; Zhang, Z.T.; Sun, Q.; Huang, Y.J. An improved ant colony algorithm for multiple unmanned aerial vehicles route planning. *J. Frankl. Inst.* **2024**, *361*, 107060. [\[CrossRef\]](#)
4. Zhou, Y.; Chen, S.J.; Li, X.P. A research review of improved ant colony algorithm. *Inf. Comput. (Theor. Ed.)* **2021**, *33*, 57–59.
5. Châari, I.; Koubaa, A.; Bennaceur, H.; Trigui, S.; Al-Shalfan, K. smartPATH: A hybrid ACO-GA algorithm for robot path planning. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation (CEC), Brisbane, QLD, Australia, 10–15 June 2012; pp. 1–8. [\[CrossRef\]](#)
6. Akopov, A.S. MBHGA: A Matrix-Based Hybrid Genetic Algorithm for Solving an Agent-Based Model of Controlled Trade Interactions. *IEEE Access* **2025**, *13*, 26843–26863. [\[CrossRef\]](#)
7. Li, X.; Wu, X.B.; Fang, T.T. A Hybrid Ant Colony Optimization Algorithm Enhanced with Mutation and Metabolism. *J. Dali Univ.* **2025**, *10*, 18–24.
8. Guo, C.C.; Tian, L.Q.; Wu, W.X. A review on the application of ant colony algorithm in solving the traveler problem. *Comput. Syst. Appl.* **2023**, *32*, 1–14. [\[CrossRef\]](#)
9. Zhang, Z.; Li, Y.; Chen, H. Improved ant colony algorithm for AGV path planning in automated container terminals. *IEEE Access* **2020**, *8*, 76348–76358.
10. Gutjahr, W.J. ACO algorithms with guaranteed convergence to the optimal solution. *Inf. Process. Lett.* **2002**, *82*, 145–153. [\[CrossRef\]](#)
11. Wu, R.H.; Lee, Y.H.; Tseng, H.W.; Jan, Y.G.; Chuang, M.H. Study of characteristics of RSSI signal. In Proceedings of the 2008 IEEE International Conference on Industrial Technology, Chengdu, China, 21–24 April 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–3.
12. Wu, Z.Z.; Wang, Y.T.; Fu, J.Q. A hybrid RSSI and AoA indoor positioning approach with adapted confidence evaluator. *Ad Hoc Netw.* **2024**, *154*, 103375. [\[CrossRef\]](#)
13. Daniş, F.S. Live RSSI filtering for indoor positioning with Bluetooth Low-Energy. In Proceedings of the 2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN), Beijing, China, 5–8 September 2022; pp. 1–8. [\[CrossRef\]](#)
14. Yang, Z.; Zhou, Z.; Liu, Y. From RSSI to CSI: Indoor localization via channel response. *ACM Comput. Surv.* **2013**, *46*, 1–32. [\[CrossRef\]](#)
15. Sadowski, S.; Spachos, P. RSSI-based indoor localization with the internet of things. *IEEE Access* **2018**, *6*, 30149–30161. [\[CrossRef\]](#)
16. Hodson, T.O. Root mean square error (RMSE) or mean absolute error (MAE): When to use them or not. *Geosci. Model Dev. Discuss.* **2022**, *2022*, 1–10. [\[CrossRef\]](#)
17. Chai, T.; Draxler, R.R. Root mean square error (RMSE) or mean absolute error (MAE)?—Arguments against avoiding RMSE in the literature. *Geosci. Model Dev.* **2014**, *7*, 1247–1250. [\[CrossRef\]](#)
18. Payton, D.; Daily, M.; Estowski, R.; Howard, M.; Lee, C. Pheromone robotics. *Auton. Robots* **2001**, *11*, 319–324. [\[CrossRef\]](#)
19. Zhang, W.; Shi, Y. Decoupled parameter control for hybrid PSO-ACO: A quantum entanglement inspired approach. In Proceedings of the ACM SIGEVO Conference on Genetic and Evolutionary Computation (GECCO), Boston, MA, USA, 9–13 July 2022; pp. 234–241.
20. Joe, P.I.R. A hybrid PSO-ACO algorithm to facilitate software project scheduling. *Int. J. Eng. Comput.* **2022**, *18*, 1–12. [\[CrossRef\]](#)
21. Phoemphon, S.; Leelathakul, N.; So-In, C. An enhanced node segmentation and distance estimation scheme with a reduced search space boundary and improved PSO for obstacle-aware wireless sensor network localization. *J. Netw. Comput. Appl.* **2024**, *221*, 103783. [\[CrossRef\]](#)
22. Yiu, S.; Dashti, M.; Claussen, H.; Perez-Cruz, F. Wireless RSSI fingerprinting localization. *Signal Process.* **2017**, *131*, 235–244. [\[CrossRef\]](#)

23. Yu, X.; Chen, W.N.; Gu, T.; Yuan, H.; Zhang, H.; Zhang, J. ACO-A*: Ant colony optimization plus A* for 3-D traveling in environments with dense obstacles. *IEEE Trans. Evol. Comput.* **2018**, *23*, 617–631. [[CrossRef](#)]
24. Huang, B.; Zhan, F.; Sun, Q.; Zhang, H. A distributed robust economic dispatch strategy for integrated energy system. *IEEE Trans. Ind. Inform.* **2022**, *18*, 880–890. [[CrossRef](#)]
25. Janczak, D.; Walendziuk, W.; Sadowski, M.; Zankiewicz, A.; Konopko, K.; Slowik, M.; Gulewicz, M. Indoor positioning based on Bluetooth RSSI mean estimator for the evacuation supervision system application. *IEEE Access* **2024**, *12*, 5584–5585. [[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.