BLE-Based Outdoor Localization With Two-Ray Ground-Reflection Model Using Optimization Algorithms

Sun, Yu

Finnerty, Patrick

Ohta, Chikara

(Citation)
IEEE Access,12:45164-45175

(Issue Date)
2024-03-22

(Resource Type)
journal article

(Version)
Version of Record

(URL)
https://hdl.handle.net/20.500.14094/0100488593

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 2 February 2024, accepted 13 March 2024, date of publication 22 March 2024, date of current version 29 March 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3380897

RESEARCH ARTICLE

BLE-Based Outdoor Localization With Two-Ray Ground-Reflection Model Using Optimization Algorithms

YU SUN, (Student Member, IEEE), PATRICK FINNERTY, (Member, IEEE), AND CHIKARA OHTA, (Member, IEEE)
Graduate School of System Informatics, Kobe University, Kobe 657-8501, Japan
Corresponding author: Yu Sun (yu.sun@fine.cs.kobe-u.ac.jp)

This work was supported by Japan Society for the Promotion of Science (JSPS) KAKENHI under Grant JP21H04914.

ABSTRACT In outdoor positioning, the global positioning system (GPS) is currently the most commonly used method. Considering the power consumption required to use GPS, it may not be efficient for tracking in smaller areas, such as outdoor grazing areas in Japan, and using wireless sensor networks seems more feasible. There are several methods for BLE-based positioning. Because the angles of arrival (AoA) and time of flight (ToF) require additional equipment, the RSSI-based localization method is the most cost-efficient. Owing to the outdoor environment, the RSSI transmission model follows a two-ray ground-reflection model, this can lead to large errors in the trilateration positioning method. On the other hand, fingerprint positioning requires the creation and maintenance of a large database. This paper proposes an RSSI-based positioning method formulated as an optimization problem. We evaluated the performance of various algorithms in two application scenarios. Our simulation results show that a high localization accuracy can be obtained using this localization method. In contrast to other methods, this approach does not necessitate the use of supplementary equipment, as is the case with AoA and ToF, nor does it require the establishment and upkeep of an RSSI fingerprint database.

INDEX TERMS BLE, firefly algorithm, localization, simulated annealing, wireless sensor networks.

I. INTRODUCTION

In recent years, the demand for improving outdoor positioning accuracy has been increasing. Whether it is global navigation satellite system (GNSS) based positioning or wireless sensor network (WSN) based positioning, it is necessary to provide high-precision outdoor positioning services as people's requirements for the quality of IoT services are constantly increasing.

GNSS is widely used in outdoor positioning because it provides navigation and positioning services on a global scale without geographical restrictions [1]. However, depending on the usage scenarios, WSN also has its own advantages. For example, WSNs have strong deployment flexibility and can operate in places where GNSS coverage is unreliable. Several WSNs have been designed for low-power applications. Compared to GNSS receivers which need to receive satellite signals continuously, WSN nodes can sleep as needed to save energy. WSNs can operate in a closed private network to provide users with higher data security and privacy protection [2]. Based on these factors, research on WSN-based positioning technologies is necessary.

In a previous study, the location information of Japanese Wagyu cattle was used to predict estrus [3]. Indeed, when cattle are in heat, they tend to chase other cattle and try to mount them. With a positioning system that achieves a margin of error of 1 meter or less, we can accurately predict the estrus status of cattle based on their behavior. In this previous study, the field considered was 144 meters long, 88 meters wide, and surrounded by 20 antennas. Each cow was equipped with a collar fitted with four Bluetooth low energy (BLE) tags. A fingerprint method is used with a long short-term

The associate editor coordinating the review of this manuscript and approving it for publication was Nadeem Ahmed.

memory (LSTM) neural network. The positioning accuracy error result obtained using this method was 5.25 meters on average. Considering the covered area, the localization accuracy achieved was quite good. However, this is not sufficient for estrus prediction.

For most of the current research, much focus has been placed on improving the existing localization methods. For example, using a Kalman filter in the angle of arrival (AoA) localization method [4], and using multiple localization methods or sensors of various natures [5]. However, most studies have focused on indoor localization. In indoor localization, the log-normal transmission model is commonly employed as the propagation model because of its effectiveness in capturing signal behavior in confined spaces. However, this model is not typically suited for outdoor localization scenarios, where different factors such as open space propagation and environmental variables come into play. The two-ray ground-reflection model is commonly used for outdoor localization. This model effectively accounts for signal reflections from the ground and other surfaces prevalent in outdoor environments, providing an accurate representation of signal propagation in open areas.

This study proposes a new outdoor localization method that is formulated as an optimization problem. BLE was chosen as the transmission technology to provide received signal strength indicator (RSSI) data. We transformed the localization problem into an optimization problem by minimizing the RMSE of the measured RSSI of the target location over the estimated RSSI. Regarding the optimization algorithms employed to solve this optimization problem, we considered the genetic algorithm (GA), firefly algorithm (FA), particle swarm optimization (PSO), simulated annealing (SA), gradient descent (GD), and memetic firefly algorithm simulated annealing (MFASA). The two-ray ground-reflection model was used as the transmission model, as it is the most realistic for outdoor settings.

The primary focus of this study is to evaluate the performance of our localization method in a simulated environment. We designed a general simulation area of 100 meters by 100 meters to evaluate the localization error under different conditions. This approach allowed us to focus on the fundamental aspects of our localization method, providing clear insights into its feasibility and performance in reducing localization errors. The contributions of this study are as follows:

1) We formulate outdoor localization as an optimization problem in which the objective is to minimize the difference between the measured and simulated RSSI at each anchor node.
2) We evaluate the performance of the six algorithms on this challenging problem to determine the most suitable one for such a scenario.

The remainder of this paper is organized as follows. Section II introduces the work related to current studies on outdoor and indoor localization. Section III describes the environment and defines the optimization problem. Section IV introduces the algorithms used in this study. Section V presents the simulation results and evaluation of all algorithms in different scenarios. We conclude in Section VI.

II. RELATED WORK

In this section, we introduce the current BLE-based localization method and propagation model for an outdoor environment.

Bluetooth low energy (BLE) is a widely used short-range wireless transmission technology that can support most cell phones and computers. It has low power consumption and high security, making it suitable for indoor and outdoor applications. BLE positioning methods include RSSI-based, time-of-flight-based, angle-based, and fingerprint-based positioning [2].

Localization based on RSSI involves measuring the signal strength between the base station (BS) and target sensor and then calculating the distance between them using a propagation model [6]. The location of the target sensor is determined using the trilateration method. This positioning method has an accuracy of approximately 2 to 4 meters [7].

Time of Flight (ToA) based localization methods include Time of Arrival (ToA) and Time Difference of Arrival (TDoA) [8]. ToA is similar to RSSI in that it calculates the distance between the BS and target sensor using time instead of RSSI, and the target position is then determined using the trilateral localization method. This method can provide high localization accuracy. However, this method requires precise clock synchronization owing to the short signal propagation time. Without precise clock synchronization, large positioning errors can occur. The TDoA calculates the relative distance between the BS and target node by calculating the time difference between the signal arrivals. The TDoA can also provide high localization accuracy. In contrast to the ToA, the TDoA does not require clock synchronization between the receiver and transmitter. However, achieving time synchronization between multiple receivers remains a challenge for this method [9].

The AoA is a localization method that calculates the angle between the target sensor and the fixed access point (AP) with respect to the North Pole. It usually requires more than two APs and has a high positioning accuracy. However, it requires the use of an antenna capable of measuring the angle [4], [10].

Fingerprint-based localization utilizes the spatial differences of RSSI in different locations. The RSSI at a specific place in space is used as the fingerprint of the location, and a location-fingerprint relation database is established to realize the estimation of the user's location through fingerprint matching [11].

The RSSI, which measures the strength of the signal, is used in location because of its simplicity, as it does not require additional equipment such as other methods [2]. Unlike the time-of-flight method, which requires high-precision clocks, or the AoA method, which requires

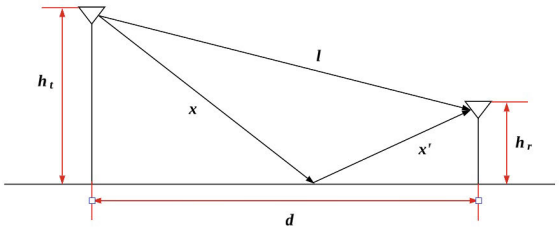


FIGURE 1. Two-ray ground-reflection model [12].

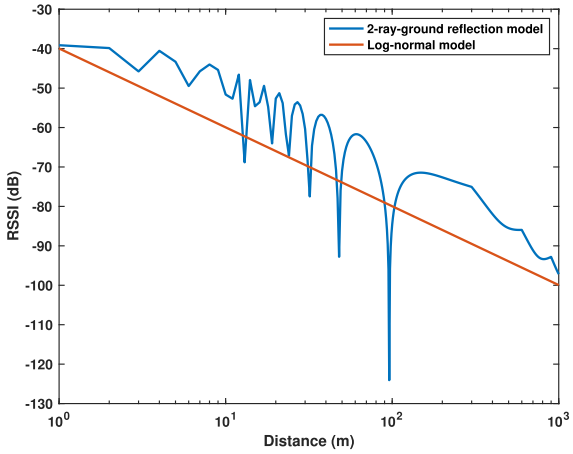


FIGURE 2. Propagation of two-ray reflection-model and log-normal model.

an antenna array, the RSSI provides a more straightforward approach. In addition, the fact RSSI decreases with distance makes it a practical choice for proximity-based applications.

In outdoor positioning, because it is an open area, the general log-normal model is not applicable, and a two-ray ground-reflection model is required. The two-ray ground-reflection model is a commonly used signal propagation model in wireless communications that considers the interaction between the direct and ground reflection paths. In this model, the signal between the transmitting and receiving antennas follows two paths: a direct path and a path that is reflected by the ground as illustrated in Fig. 1 [12]. The difficulty with using the two-ray ground-reflection model is that the conversion between RSSI and distance is not a one-to-one correspondence, as is the case with the log-normal model. In other words, the distance can be directly converted into RSSI using the model, but each RSSI value can correspond to multiple distances as shown in Fig. 2.

When using the two-ray ground-reflection model, determining the target position coordinates based on the RSSI becomes much more difficult, which is a notable difficulty in traditional positioning approaches.

Using a fingerprint method can effectively avoid the problem that the RSSI value cannot be converted into distance. However, RSSI fingerprinting presents significant challenges, particularly in terms of building and maintaining a comprehensive fingerprint database. In large areas, the exhaustive collection of fingerprint data is a demanding task, often limited by logistical and practical constraints. This complexity underscores the limitations of the method in large-scale environments.

Although ToF, AoA, and fingerprinting all provide high accuracy, RSSI does not require additional measurement equipment or maintenance of large fingerprint databases. Thus, this is the most cost-effective approach. Owing to the properties of the two-beam ground reflection model, the distance cannot be calculated directly from the measured RSSI. Trilateration cannot be used for positioning. Therefore, we decided to localize by utilizing an optimization algorithm to avoid the error-prone distance calculations resulting from the use of the two-ray ground-reflection model.

III. ENVIRONMENT AND PROBLEM FORMULATION

In this section, we introduce our environment model and objective function. Transforming a localization problem into an optimization problem can avoid the conversion of the RSSI and distance. We plan to localize using algorithms such as particle swarm optimization (PSO) or firefly algorithm (FA) to minimize the difference between the measured RSSI and estimated RSSI between the target position and anchor nodes.

A. ENVIRONMENT MODEL

In this study, we consider a 100-meter wide square field surrounded by eight anchor nodes to remain close to the real situation presented in [3]. Considering that the usage environment is outdoor, we choose to use the 2.4 GHz frequency because of its longer transmission distance compared to 5 GHz. The height of the transmitter antennas was 4 meters, and the height of the receiver antennas (i.e., the cow necklaces) was 1.5 meters. The detailed environmental parameters are summarized in Table 1, and Fig. 3 shows the simulation area and anchor nodes layout around the field.

B. OPTIMIZATION PROBLEM FORMULATION

As explained in Section II, because of the characterization of the two-ray ground-reflection model, we cannot use the RSSI value to calculate the distance from the anchor nodes directly However, the RSSI value corresponding to a distance is unique, which means that the received RSSI value can be predicted based on the antenna's distance from the anchor node. Therefore, we can transform this localization problem into an optimization problem. In a two-dimensional search space (corresponding to the field at hand), the problem involves finding the point at which the RSSI computed according to the two-way ground-reflection model is as close as possible to the RSSI measured at each anchor node.

The distance between the anchor node and the estimated position can be calculated using the Euclidean distance

$$d_{\text{eu}} = \sqrt{(a_2 - a_1)^2 + (b_2 - b_1)^2}, \tag{1}$$

where

- d_{eu} is the Euclidean distance,
- (a_1, b_1) are the coordinates of the first point, and
- (a_2, b_2) are the coordinates of the second point.

TABLE 1. Environment parameters.

Parameter	Value
Area Length	100 meters
Area Width	100 meters
Anchor Nodes (coordinates)	(0, 0), (50, 0), (100, 0), (100, 50), (100, 100), (50, 100), (0, 100), (50, 100)
Frequency	2.4 GHz
Transmitter Antenna Height (h_t)	4 meters
Receiver Antenna Height (h_r)	1.5 meters

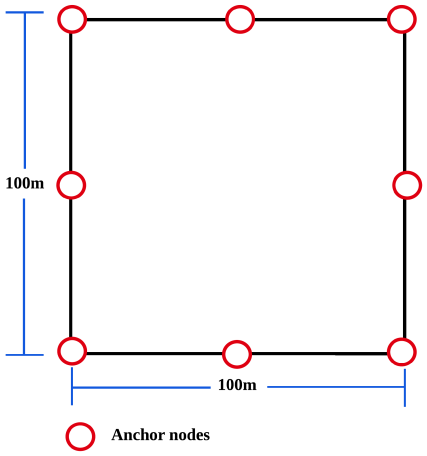


FIGURE 3. Simulation area and anchor nodes location.

Using the distance calculated above, the estimated power received from the antenna at each anchor node can be calculated using the two-ray ground-reflection model [12].

The received signal power can be calculated by

$$P_{\text{re}} = P_{\text{tr}} \left(\frac{\lambda}{4\pi}\right)^2 \left| \frac{\sqrt{G_{\text{tr}}}}{l} + \frac{R\sqrt{G_{\text{re}}} e^{-j\Delta\phi}}{x + x'} \right|^2, \tag{2}$$

where

- P_{re} is the received power,
- P_{tr} is the transmitted power,
- λ is the wavelength,
- G_{tr} is the gain at the transmitter,
- l is the distance from the transmitter to the receiver,
- G_{re} is the gain at the receiver,
- R is the reflection coefficient,
- j is the imaginary unit,
- $\Delta\phi$ is the phase shift based on path differences,
- x is the distance from the transmitter to the reflection point, and
- x' is the distance from the reflection point to the receiver.

The value of $x + x'$ can be calculated by

$$x + x' = \sqrt{(h_{\text{t}} + h_{\text{r}})^2 + d^2}, \tag{3}$$

and the value of l can be calculated by

$$l = \sqrt{(h_{\text{t}} - h_{\text{r}})^2 + d^2}, \tag{4}$$

where

- d is the plane distance between transmitter and receiver,
- h_{t} is the height of transmitter, and
- h_{r} is the height of receiver.

$\Delta\phi$ can be calculated by

$$\Delta\phi = \frac{2\pi(x + x' - l)}{\lambda}. \tag{5}$$

The estimated RSSI can be derived from the received power with

$$RSSI_i^{\text{est}} = 10 \log_{10} P_{\text{re}}, \tag{6}$$

where $RSSI_i^{\text{est}}$ is the estimated RSSI at target position. Based on the above formulas, we can determine the estimated RSSI value from any position in the region to the anchor nodes. Thus, minimizing the difference between the measured and estimated RSSI values makes it possible to determine the target position. The following objective function was employed:

$$f = \sqrt{\sum_{i=1}^{n} \left(RSSI_i^{\text{measured}} - RSSI_i^{\text{est}}\right)^2}, \tag{7}$$

where

- n is the number of anchor nodes, and
- $RSSI_i^{\text{measured}}$ is the measured RSSI at target node i.

The objective value is the RMSE of the difference between the measured and estimated RSSI value for each antenna. The smaller the objective value, the closer the position to the target node.

C. DISCUSSION

In our simulations, we use the "twoRayChannel" toolbox from the MATLAB Radar Toolbox as the transmission model to simulate the transmission signal and generate the measured RSSI values. The system then employs the equation model detailed in Section III-B to calculate the estimated RSSI. Fig. 4 shows the relationship between the RSSI and the distance in the MATLAB and the equation models.

Both models are close but not identical. The difference between the two models can result in localization inaccuracies. Furthermore, particular environmental factors in real settings may also lead to errors, which will be discussed in Section V.

The topology of the objective function when the target node was located at (50, 50) is shown in Fig. 5. The environmental parameters were the same as those listed in Table 1, and the RSSI between each anchor node and target position were generated using Matlab's twoRayChannel.

Upon inspection of the topology, it became apparent that the functional surface revealed small bumps, valleys,

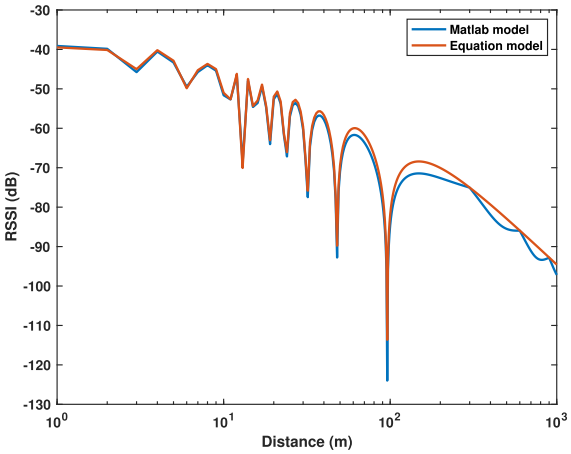


FIGURE 4. MATLAB model and equation model comparison.

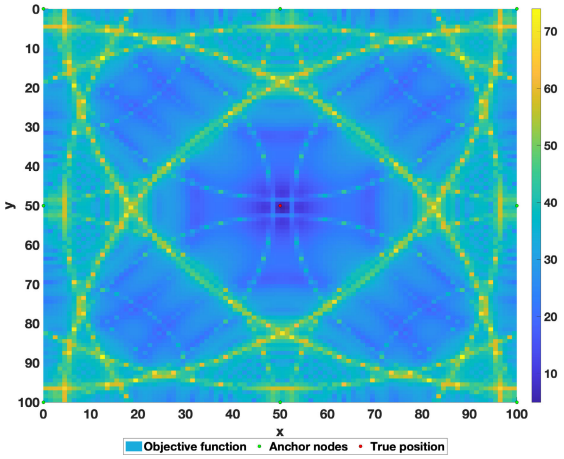


FIGURE 5. Objective function topology when the target node is located in the center of the field at coordinates (50, 50).

and irregularities. Such a topology presents a challenge for simple optimization algorithms such as gradient descent to compute the global optimal solution.

IV. OVERVIEW OF ALGORITHMS CONSIDERED

In this section, we introduce the algorithms used in this study. As outlined in Section III, we convert the localization problem into an optimization problem. Many algorithms can tackle such problems, including the gradient descent (GD), simulated annealing (SA), particle swarm algorithms (PSO), and genetic algorithms (GA). We employed several optimization algorithms, including the firefly algorithm (FA), SA, memetic firefly algorithm simulated annealing (MFASA), GD, PSO, and GA. MFASA is a hybrid algorithm that combines the features of FA and SA. As presented in [13], this algorithm is adept at solving issues related to the local optima. Given the complexity of the problem at hand, simple and widely used algorithms such as gradient descent often tend to converge on a local optimal solution rather than a global optimal solution. Therefore, we included MFASA for comparison with the other algorithms to address this specific challenge.

A. FIREFLY ALGORITHM (FA)

The FA is a particle swarm-like algorithm proposed by Yang [14]. The FA is inspired by the flashing behavior of fireflies, which is an intelligent random algorithm for global optimization problems. The basic steps of FA are as follows:

1) INITIALIZATION

Randomly initialize a set of fireflies within the search space that represents potential solutions. Each firefly is assigned an intensity proportional to the value of the objective function. The relationship between intensity I and objective function f for a *minimization problem* can be expressed as

$$I(x) = -f(x), \tag{8}$$

where x represents a firefly in the search space.

2) MOVING FIREFLIES

Each firefly moves toward fireflies that are brighter than them (the brightest firefly of the whole swarm does not move). The extent to which a firefly moves toward a brighter congener is determined by the distance between them according to the function β:

$$\beta(r) = \beta_0 e^{-\gamma r^2}, \tag{9}$$

where

- r is the distance between the fireflies,
- β_0 is a positive constant, and
- γ is the light absorption coefficient.

3) BRIGHTNESS UPDATE

After each firefly in the swarm has moved, its brightness is updated based on its new position according to (8).

These last two steps are repeated until a predefined number of iterations is reached or a predefined objective value is reached. The brightest firefly of the last iteration is the best solution.

B. SIMULATED ANNEALING (SA)

SA was independently proposed by multiple authors [15], [16]. It is a stochastic optimization algorithm based on the Monte Carlo method. The inspiration comes from the annealing process in physics and can be used to solve optimization problems. The SA starts from a relatively high initial temperature, and with a continuous decrease in the temperature, it randomly finds the global optimal solution in the search space. The annealing process starts with the initial solution s. In each iteration, a new solution s' is generated in the neighborhood of the current solution. The basic steps of SA are as follows:

1) ENERGY DIFFERENCE

Whenever a new solution is considered, the energy difference from the current solution is calculated as follows:

$$\Delta E = f(s') - f(s), \tag{10}$$

where

- ΔE is the energy difference between the current solution and the newly generated solution (difference in objective value between the two solutions),
- f is the objective function,
- s is the current solution, and
- s' is the new solution generated in the neighborhood of s.

2) ACCEPTANCE CRITERION

The acceptance or rejection of a new solution is based on two main conditions:

If the new solution is better than the current solution ($\Delta E < 0$), it is directly accepted. An inferior new solution ($\Delta E \geq 0$), it is accepted according to the probability based on the current temperature. The following equation calculates this probability

$$P_{\text{accept}} = e^{-\Delta E/T}, \tag{11}$$

where

- T is the current temperature, and
- P_{accept} is the acceptance probability.

A random number $r \sim U(0, 1)$ is generated and compared with the above probabilities to determine whether to accept the new solution. If $r < P_{\text{accept}}$, the algorithm accepts it.

3) TEMPERATURE DISPATCH

The high temperature at the beginning of the algorithm encourages an extensive search, and new solutions may be accepted even if they are poor. Over time, the temperature gradually decreased, thus reducing the chance of accepting poor-quality solutions. The temperature dispatch is determined by

$$T_n = T_0 \times \rho^n, \tag{12}$$

where

- T_n is the current temperature,
- T_0 is the initial temperature, and
- ρ is the cooling rate.

4) TERMINATION CONDITIONS

Once the temperature is reduced to a predetermined threshold or the algorithm performs a predetermined maximum number of iterations, it terminates and returns the current best solution.

C. MEMETIC FIREFLY ALGORITHM SIMULATED ANNEALING (MFASA)

MFASA was proposed by Nadia and Mahdi. In each iteration, after completing the FA move step, SA was performed on each solution to generate alternatives to the current fireflies. If these alternatives are better solutions than those obtained after the move step, the algorithm continues with these solutions instead. The authors claimed that the algorithm outperforms the common FA in terms of accuracy and convergence speed [13]. The pseudo-code of MFASA is presented in Algorithm 1.

Algorithm 1 MFASA

```
Initialize firefly population, objective function, and initial temperature T
while termination criteria not met do
  for each firefly i do
    for each firefly j do
      if intensity of j > intensity of i then
        Move firefly i towards j
      end if
    end for
  end for
  for each firefly i do
    Choose a new position for i in its neighborhood
    Calculate the change in objective function ΔE
    if ΔE < 0 or r < exp(−ΔE/T)) then
      Accept the new position for i
    end if
  end for
  Decrease T using cooling rate
end while
Return best solution found
```

The algorithm terminates when the predefined objective value is reached or the limit number of iterations is reached. The firefly global best at the moment the algorithm halts is considered as the solution.

Combining these two techniques increases the chances of finding the global minimum of the objective function as simulated annealing adds random perturbations to the solution candidates, allowing the fireflies to escape areas of local minima and explore other regions of the search space.

D. PARTICLE SWARM OPTIMIZATION (PSO)

PSO introduced by Kennedy and Everhartas is a population-based optimization technique inspired by the social behaviors of bird flocking [17]. The basic steps of PSO are as follows [18]:

1) INITIALIZATION

Initialize a swarm of n particles, each represented by a position vector x_i and velocity vector v_i.

2) FITNESS EVALUATION

The fitness of each particle is evaluated according to the value of the objective function $f(x_i)$.

3) UPDATE PERSONAL AND GLOBAL BEST

Each particle tracks of its personal best position $\boldsymbol{pbest}_n$ and the global best position $\boldsymbol{gbest}$ of the swarm.

4) UPDATE THE VELOCITY AND POSITION

The velocity and position of the particles in each iteration are updated by

$$\begin{aligned} \boldsymbol{v}_i^{\text{new}} &= w \times \boldsymbol{v}_i + c_1 \times r_1 \times (\boldsymbol{pbest}_i - \boldsymbol{x}_i) \\ &\quad + c_2 \times r_2 \times (\boldsymbol{gbest} - \boldsymbol{x}_i), \\ \boldsymbol{x}_i^{\text{new}} &= \boldsymbol{x}_i + \boldsymbol{v}_i^{\text{new}}, \end{aligned} \tag{13}$$

where

- w denotes the inertia weight which helps to balance global and local search capabilities,
- c_1 and c_2 are the acceleration coefficients. c_1 dominates the attraction of the particles toward $\boldsymbol{pbest}_n$, whereas c_2 controls the strength of the particle's attraction to the $\boldsymbol{gbest}$, and
- r_1 and r_2 are random numbers generated independently and uniformly in interval [0,1]. These random numbers are used to introduce randomness and help the algorithm escape the local optima.

These steps are repeated until a predefined number of iterations is reached or until some other termination condition is met. The best solution of the last iteration is then considered the solution to the optimization problem.

E. GRADIENT DESCENT (GD)

GD is an iterative method for optimizing differentiable functions introduced by Cauchy [19]. The basic steps of GD are as follows:

1) INITIALIZATION

Start with an initial point x_0 within the search space.

2) ITERATIVE UPDATE

In each iteration, update the position in the direction opposite to the gradient of the objective function as

$$x_{t+1} = x_t - \eta \cdot \nabla f(x_t), \tag{14}$$

where

- x_t is the current position in the parameter space at iteration t,
- η is the learning rate, and
- $\nabla f(x_t)$ is the gradient of the objective function at x_t.

These steps are repeated until a predefined number of iterations is reached or a predefined objective value is reached.

F. GENETIC ALGORITHM (GA)

GA is a heuristic search and optimization algorithm inspired by the natural evolutionary process [20]. It mimics the process of natural selection by selecting the most suitable individuals to reproduce in order to produce the next generation of individuals. The basic steps of GA are as follows [21]:

1) INITIALIZATION

A population of N individuals is initialized. Each individual, also known as a chromosome, is a potential solution for this problem. The chromosome can be represented in various forms, such as a 2-dimensional search area with a real value, which can be represented as

$$\text{Chromosome}_i = (a_i, b_i), \tag{15}$$

where a_i, b_i represents the coordinates of Chromosome$_i$ in a 2-dimensional area.

2) FITNESS EVALUATION

The fitness of each chromosome's fitness is evaluated based on an objective function $f(x)$, which measures the quality of the represented solution.

3) SELECTION

Selection simulates the survival of the fittest. The individuals were selected based on their fitness levels. A common method is roulette wheel selection, where the probability of selection is proportional to the fitness of the individual.

4) ELITE SELECTION

The best individual (elite) from the current population are retained directly in the next generation to ensure that the best solutions currently found are not lost.

5) CROSSOVER

Selected individuals produce offspring by crossover (pairing and mixing genetic information). This process produces new individuals that contain combinations of features from the parent generation, which helps explore the solution space as

$$\text{Child}_1, \text{Child}_2 = \text{Crossover}(\text{Parent}_1, \text{Parent}_2). \tag{16}$$

6) MUTATION

Mutation introduces genetic diversity into the population. This helps the algorithm jump out of the local optima and explore new solution space as

$$\begin{aligned} a_i' &= \begin{cases} a_i + \Delta & \text{with mutation probability } p, \\ a_i & \text{otherwise}, \end{cases} \\ b_i' &= \begin{cases} b_i + \Delta & \text{with mutation probability } p, \\ b_i & \text{otherwise}, \end{cases} \end{aligned} \tag{17}$$

where

- a_i', b_i' are the coordinates after mutation,
- a_i, b_i are the coordinates before mutation,
- Δ is used to indicate the amount of change in the a and b coordinates.

These steps are repeated until a termination condition is met, such as reaching a predetermined number of iterations or finding a satisfactory solution.

V. EVALUATION

In this section, we evaluate this proposed localization method. First, we evaluated the performance of the algorithms when confronted with random points in a field. In the second step,

TABLE 2. Optimization algorithms parameters.

Algorithm	Parameters	Value
Genetic Algorithm	Iterations	300
	Population	100
	Crossover probability	0.9
	Mutation probability	0.1
	Elitism rate	0.02
Memetic Firefly Algorithm Simulated Annealing	Population	100
	Iterations	300
	α	0.8
	β	0.4
	γ	0.9
	θ	0.9
	T	1500
	ρ	0.95
Particle Swarm Optimization	Population	100
	Iterations	300
	c_1	1.5
	c_2	1.5
	w	0.9
Gradient Descent	Iterations	300
	η	0.1
Firefly Algorithm	Population	100
	Iterations	300
	α	0.8
	γ	0.4
	β	9
	θ	0.9
Simulated Annealing	Iterations	1000
	T	100
	ρ	0.99

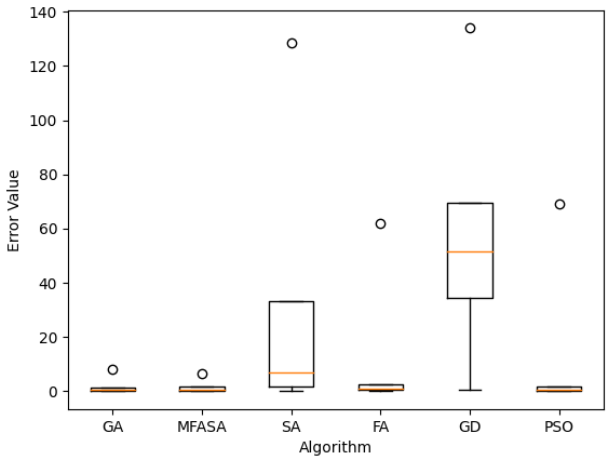


FIGURE 6. Error values of different algorithms in random position.

we generate three random waypoints and restrict the search space to the neighborhood of the previous point. Indeed, in our case, it is reasonable to assume that successive points will be close to one-another. Finally, we discuss the overall results and trends in Section V-C.

For parameter tuning, we used the same data and grid search method for each algorithm. Through a grid search, we obtained the optimal parameters for each algorithm as shown in Table 2.

A. RANDOM POSITION

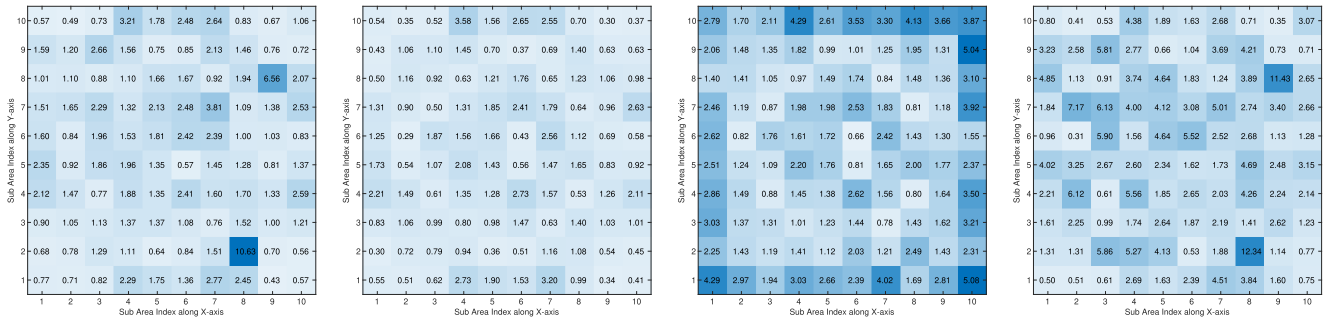
We randomly generated 50 positions within a 100 meters by 100 meters field. Each algorithm was run ten times at each position. Statistical analyses were performed based on the results obtained. The error distribution for each algorithm is shown in Fig. 6.

The performances of the various algorithms in terms of localization accuracy exhibited significant variations. MFASA demonstrated the lowest mean error at 1.10 meters, indicating a superior average localization accuracy. Its maximum error of 6.62 meters suggests a relatively stable performance even in the worst-case scenarios. GA displayed a slightly higher mean error of 1.15 meters. However, the maximum error of 8.11 meters indicates the potential for substantial deviations in certain cases. PSO and FA reported mean errors of 1.83 meters and 2.02 meters, respectively, showing overall less efficacy compared to MFASA and GA. In particular, PSO exhibited a significant maximum error of 69.15 meters, highlighting the possibility of large localization inaccuracies under some conditions. SA and GD had considerably higher mean errors of 20.44 meters and 52.89 meters, respectively, indicating markedly lower localization precision in this setup.

In summary, MFASA and GA outperformed other algorithms. In particular, MFASA excels in terms of both mean accuracy and stability. In contrast, SA and GD demonstrated subpar performance. Therefore, we chose not to consider SA and GD further in the remainder of our evaluation.

To evaluate the performance of various algorithms across different regions, we divided the 100 meters by 100 meters area into 100 subareas, each measuring 10 meters by 10 meters. Within each subarea, we generated ten location coordinates. We then ran each algorithm on these points and computed the average error for each subarea. The results are shown in Fig. 7.

The mean error for the GA is 1.58 meters, indicating that the overall error is within the control. The standard deviation of 1.24 shows some fluctuation in the error distribution, but it is relatively small. The mean error of 1.15 meters for



(a) GA (b) MFASA (c) FA (d) PSO

FIGURE 7. Average error of 4 algorithms in each subarea.

MFASA was the lowest of the four algorithms, indicating the best overall performance. The standard deviation of 0.70 is relatively small, indicating good consistency of the results. The maximum error of 3.58 meters was the smallest of the four algorithms, indicating that the performance was still good in extreme cases. For PSO, the mean error is 2.75 meters, which is significantly higher than that of the other algorithms, showing poorer overall performance. The maximum error of 12.34 meters is very high, indicating that the algorithm performance may be unstable, similar to the GA in some special areas. The mean error of FA is 1.99 meters, which is located between those of the GA and PSO. The standard deviation is 0.99, which is relatively small compared to those of GA and PSO, but larger than that of MFASA. The maximum error is 5.08 meters, which is larger than that of MFASA but smaller than those of GA and PSO.

MFASA performed best in this set of comparisons, with the lowest mean error and highest stability. The GA exhibits the next-best performance with better mean error and stability. It is important to note that it had a maximum error of 10.64 meters. FA lags behind MFASA and GA in performance but may still outperform PSO. PSO appears to have the best erratic performance, with the highest mean and maximum errors.

Regarding the location of outliers, no significant outliers appeared in the FA and MFASA results. In contrast, PSO and GA had the same outliers at locations (2, 8) and (8, 7). It is impossible to determine whether the specificity of the region is responsible for this result or if there is another reason.

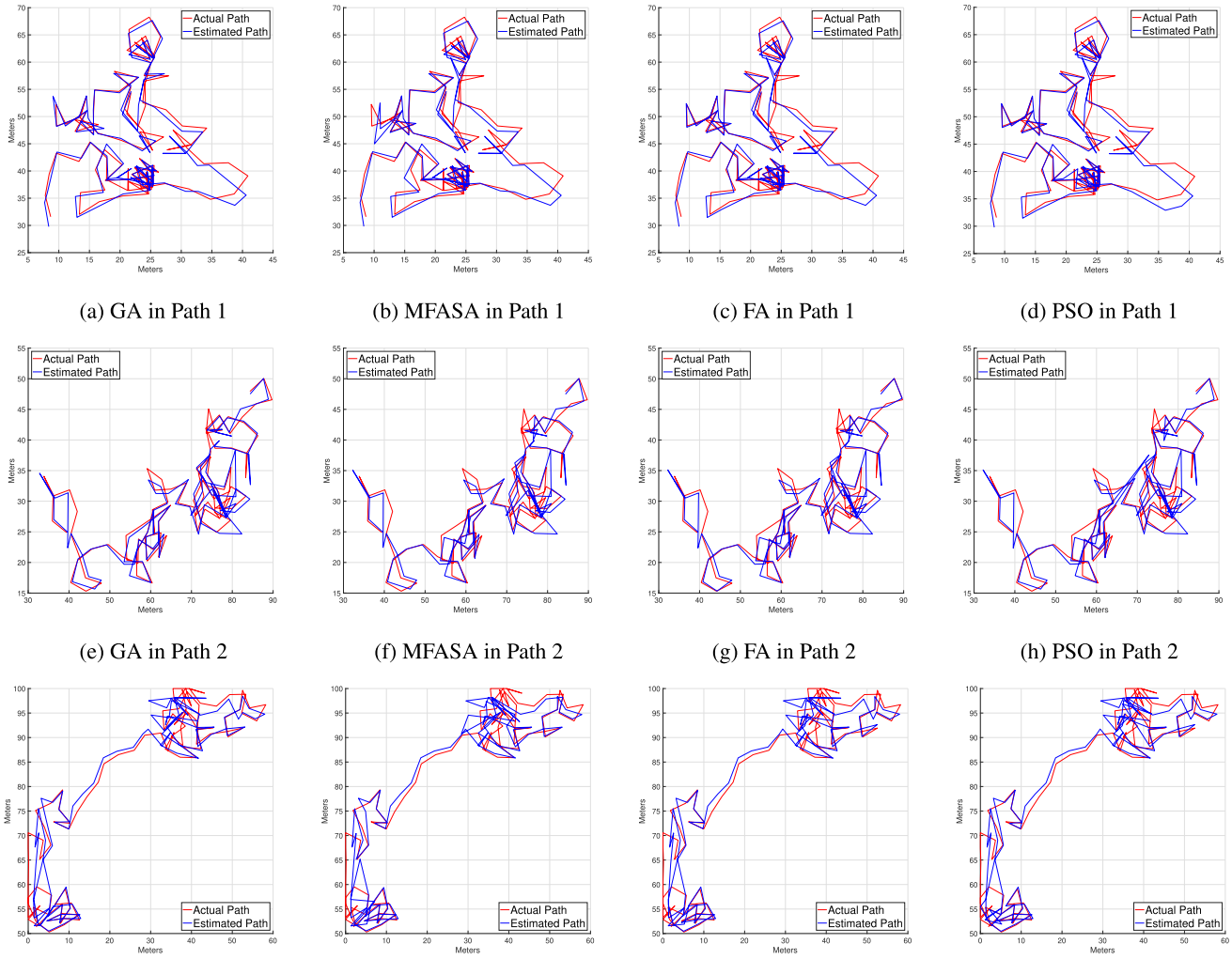
Overall, the errors for GA, MFASA, and FA were evenly distributed, with most of the high error regions located at the edges of the overall region. However, the GA has two outliers, and all three algorithms have low center errors. We believe that this is related to modeling errors. When the target is at the edge position, more than half or half of the anchor nodes are too far away from the target position, resulting in modeling errors.

In contrast, the PSO results did not have high errors at the edge locations and low errors at the center. Instead, large errors occur near the edge region, which is different from the results of the other three algorithms.

B. RANDOM WAYPOINT

In a real environment, neither humans nor animals move quickly. Therefore, to make the evaluation more reasonable, we decided that the first iteration of the algorithm in localization would perform a full-area search. Each subsequent iteration redefines the region in which the algorithm is executed based on the localization results of the previous iteration. Based on the results of the previous set, we decided to define the extent of this new search area as a circle of radius 20 meters centered on the location coordinates of the previous localization result. When the cattle run, it can reach an average speed of 7.6 meters per second. However, considering that most of the time, cattle stay somewhere for a long time or make very slow movements, we decided to set the distance between two successive waypoints to 4 meters. We randomly generated three paths, each with 100 waypoints and a distance of 4 meters between each waypoint. The results are shown in Fig. 8 and Table 3.

All four algorithms tracked the movement trajectory of the target accurately. In Path 1, both GA and PSO outperformed FA and MFASA with a mean error of 0.78 meters, and median errors of 0.59 meters and 0.51 meters, respectively. The maximum error of the GA is 3.55 meters, which is significantly lower than the maximum errors of the other algorithms. In Path 2, GA continues to perform better with a mean error of 0.89 meters and a median error of 0.59 meters. Although PSO has the highest maximum error (11.43 meters) at this waypoint, its mean and median errors (1.08 meters and 0.64 meters, respectively) still show better overall performance. In Path 3, the errors of the four algorithms are significantly higher. The average error for both FA and GA is 1.52 meters, which is the lowest among the four algorithms. The median errors were 0.89 meters and 0.83 meters, respectively, which performed well relative to the other methods. The four algorithms had similar maximum errors, with no significant differences in performance. This is due to the path that goes into a place on the field in which all



(a) GA in Path 1 (b) MFASA in Path 1 (c) FA in Path 1 (d) PSO in Path 1

(e) GA in Path 2 (f) MFASA in Path 2 (g) FA in Path 2 (h) PSO in Path 2

(i) GA in Path 3 (j) MFASA in Path 3 (k) FA in Path 3 (l) PSO in Path 3

FIGURE 8. Estimated path obtained from four optimization algorithms on three random waypoint paths.

TABLE 3. Error statistics for each waypoint.

Waypoint	Method	Mean Error	Median Error	Min Error	Max Error
1	Firefly Algorithm	0.81	0.54	0.06	6.28
	Genetic Algorithm	0.78	0.59	0.01	3.55
	Memetic Firefly Algorithm Simulated Annealing	0.88	0.62	0.06	6.37
	Particle Swarm Optimization	0.78	0.51	0.06	6.30
2	Firefly Algorithm	0.96	0.56	0.04	7.15
	Genetic Algorithm	0.89	0.59	0.07	7.16
	Memetic Firefly Algorithm Simulated Annealing	0.98	0.60	0.04	7.15
	Particle Swarm Optimization	1.08	0.64	0.04	11.44
3	Firefly Algorithm	1.52	0.89	0.05	6.53
	Genetic Algorithm	1.52	0.83	0.02	6.54
	Memetic Firefly Algorithm Simulated Annealing	1.68	0.97	0.01	7.15
	Particle Swarm Optimization	1.55	0.92	0.05	7.89

algorithms do not perform as well as in other areas (as shown in Fig. 7).

The GA had the lowest mean and median errors for all three paths, indicating a superior overall performance. Although PSO performs better in Path 1, it exhibits higher errors in Paths 2 and 3. A maximum error of 11.44 meters occurs in Path 2, which is unsatisfactory. In contrast, FA was more stable in all three paths. Although its overall performance is

not as good as that of GA, it can provide stable positioning results with acceptable errors.

The results for MFASA are quite surprising. MFASA performed the best in both the random location and subarea. However, after narrowing down the search range to a random waypoint, MFASA did not perform as well as the other algorithms. This could be due to the fact that MFASA uses the simulated annealing mechanism which causes the algorithm to accept worse results at runtime in order to speculatively avoid suboptimality. However, the limited number of searches reduces the likelihood of encountering this issue. Therefore, it is reasonable for this algorithm to present worse results than other algorithms in this setting.

C. DISCUSSION

Overall, several algorithms can provide improved localization results in this mode. In more detailed comparisons, the GA exhibits better stability and lower localization errors in this system and performs well in both the subarea and the random waypoint. FA has a higher error rate than GA. However, it consistently provides higher localization results. In the whole-field search, MFASA outperformed GA because of its strong ability to handle local optimization. However, MFASA performs poorly at random waypoints, and its localization accuracy is not as good as that of other algorithms once the search space has been somewhat restricted around the previously known location. PSO can also provide more accurate localization results but is less stable than GA, FA, and MFASA. Regarding GD and SA, these two algorithms cannot achieve highly accurate localization results in this mode. Owing to the complexity of the problem, these algorithms may fall into local optima, resulting in large errors in the localization results.

However, the present analysis is based on a simulation. In the actual environment, the impact of environmental factors on the RSSI will not be the same as that in the ideal model, and other factors, such as signal interference, also need to be considered. A previous study indicated that heavy rain can disrupt 2.4 GHz Wi-Fi signals, leading to interruptions [22]. Additionally, it is important to consider that humidity and temperature changes can cause RSSI fluctuations in real-world environments, with variations of up to ± 5 dBm. Given these factors, it is essential to verify the impact of these specific extreme conditions on WiFi signal integrity by empirically evaluating them in actual situations. We leave this to future work.

VI. CONCLUSION

This study presents a new RSSI-based outdoor localization method. The method transforms the localization problem into an optimization problem using an optimization algorithm to minimize the RMSE between the measured RSSI value and the predicted RSSI value to obtain the coordinates of the target location for localization. All simulations were based on an outdoor environment and used a two-ray ground-reflection model as the transmission model.

Overall, in evaluating different optimization algorithms in random positions, the SA and GD performed poorly. The GA, MFASA, FA, and PSO can obtain satisfactory localization errors. Among them, the smallest positioning result error of 1.1 meters was obtained using MFASA. In contrast, MFASA performs the highest positioning result in random waypoints but GA has the best performance.

Considering that different scenes have different propagation models, we believe that by replacing the propagation model, this method can also be applied to solve the problem of indoor localization or localization in other scenes. In future studies, we will use real data using this method to evaluate the practicality of the scheme. In addition, considering the influence of the environment on the RSSI, the positioning accuracy and stability of the system may need to be further improved by using fusion with other sensors or positioning techniques.

REFERENCES

[1] G. Wang, X. Xu, Y. Yao, and J. Tong, "A novel BPNN-based method to overcome the GPS outages for INS/GPS system," *IEEE Access*, vol. 7, pp. 82134–82143, 2019.
[2] S. M. Asaad and H. S. Maghdid, "A comprehensive review of indoor/outdoor localization solutions in IoT era: Research challenges and future perspectives," *Comput. Netw.*, vol. 212, Jul. 2022, Art. no. 109041.
[3] T. Yamanishi, T. Jikyo, T. Kamada, R. Nishide, C. Ohta, K. Oyama, and T. Ohkawa, "A study on outdoor localization method by recurrent deep learning based on time series of received signal strength from low power wireless tag," *IEICE Commun. Exp.*, vol. 8, no. 12, pp. 572–577, 2019.
[4] C. Guangqian, L. Zuowei, W. Xiaohong, S. Shiying, and H. Jianwen, "Research on TDOA/AOA hybrid positioning system based on Kalman filtering," in *Proc. 2nd Int. Conf. Signal Process. Syst.*, vol. 2, Jul. 2010, pp. 720–723.
[5] D. Vicente, S. Tomic, M. Beko, R. Dinis, M. Tuba, and N. Bacanin, "Kalman filter for target tracking using coupled RSS and AoA measurements," in *Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2017, pp. 2004–2008.
[6] T. Li, S. Ai, S. Tateno, and Y. Hachiya, "Comparison of multilateration methods using RSSI for indoor positioning system," in *Proc. 58th Annu. Conf. Soc. Instrum. Control Engineers Jpn. (SICE)*, Sep. 2019, pp. 371–375.
[7] A. A. Sohan, M. Ali, F. Fairooz, A. I. Rahman, A. Chakrabarty, and M. R. Kabir, "Indoor positioning techniques using RSSI from wireless devices," in *Proc. 22nd Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2019, pp. 1–6.
[8] T. Wang, H. Ding, H. Xiong, and L. Zheng, "A compensated multi-anchors TOF-based localization algorithm for asynchronous wireless sensor networks," *IEEE Access*, vol. 7, pp. 64162–64176, 2019.
[9] Y. Zhao, Z. Li, B. Hao, P. Wan, and L. Wang, "How to select the best sensors for TDOA and TDOA/AOA localization?" *China Commun.*, vol. 16, no. 2, pp. 134–145, Feb. 2019.
[10] N. A. A. Elhag, I. M. Osman, A. A. Yassin, and T. B. Ahmed, "Angle of arrival estimation in smart antenna using MUSIC method for wideband wireless communication," in *Proc. Int. Conf. Comput., Electr. Electron. Eng. (ICCEEE)*, Aug. 2013, pp. 69–73.
[11] Q. Chen and B. Wang, "FinCCM: Fingerprint crowdsourcing, clustering and matching for indoor subarea localization," *IEEE Wireless Commun. Lett.*, vol. 4, no. 6, pp. 677–680, Dec. 2015.
[12] A. Goldsmith, *Wireless Communications*. Cambridge, U.K.: Cambridge Univ. Press, 2005.
[13] N. Nekouie and M. Yaghoobi, "MFASA: A new memetic firefly algorithm based on simulated annealing," *Int. J. Mechatronics Manuf. Syst.*, vol. 5, pp. 2347–2354, Jul. 2015.
[14] X. S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, p. 78, 2010.
[15] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, May 1983.

[16] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *J. Optim. Theory Appl.*, vol. 45, no. 1, pp. 41–51, Jan. 1985.
[17] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE ICNN*, vol. 4, Nov. 1995, pp. 1942–1948.
[18] M. Clerc and J. Kennedy, "The particle swarm–explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.
[19] A.-L. Cauchy, *Méthode Générale Pour la Résolution des Systèmes D'équations Simultanées*. 1847.
[20] J. McCall, "Genetic algorithms for modelling and optimisation," *J. Comput. Appl. Math.*, vol. 184, no. 1, pp. 205–222, Dec. 2005.
[21] C. Ferreira, "Gene expression programming: A new adaptive algorithm for solving problems," *Complex Syst.*, vol. 13, no. 2, pp. 87–129, Dec. 2001.
[22] J. Bauer and N. Aschenbruck, "Towards a low-cost RSSI-based crop monitoring," *ACM Trans. Internet Things*, vol. 1, no. 4, pp. 1–26, Nov. 2020.

YU SUN (Student Member, IEEE) was born in Xi'an, China, in 1993. He received the B.S. degree in telecommunication engineering from Xi'an University of Posts & Telecommunications, China, in 2015, and the M.S. degree in telecommunication engineering from the University of Technology Sydney, Australia, in 2019. He is currently pursuing the Ph.D. degree with the Graduate School of System Informatics, Kobe University.

PATRICK FINNERTY (Member, IEEE) was born in Cholet, France, in 1995. He received the French Engineering degree in computer science and information technology from INSA Lyon, in 2018, and the Ph.D. (Eng.) degree from Kobe University, Japan, in 2022. Since February 2022, he has been an Assistant Professor with the Graduate School of System Informatics, Kobe University. His research interests include parallel and distributed computing techniques and distributed systems. He is a member of IPSJ, IEEE Computer Society, and ACM.

CHIKARA OHTA (Member, IEEE) was born in Osaka, Japan, in 1967. He received the B.E., M.E., and Ph.D. (Eng.) degrees in communication engineering from Osaka University, Osaka, in 1990, 1992, and 1995, respectively. In April 1995, he was an Assistant Professor with the Department of Computer Science, Faculty of Engineering, Gunma University. In October 1996, he was a Lecturer with the Department of Information Science and Intelligent Systems, Faculty of Engineering, University of Tokushima, where he was also an Associate Professor, in March 2001. In November 2002, he was an Associate Professor with the Department of Computer and Systems Engineering, Faculty of Engineering, Kobe University, where he was an Associate Professor with the Graduate School of System Informatics, in April 2010, and a Professor, in January 2015. In April 2016, he was a Professor with the Graduate School of Science, Technology and Innovation, Kobe University, where he has been a Professor with the Graduate School of System Informatics, since April 2022. From March 2003 to February 2004, he was a Visiting Scholar with the University of Massachusetts Amherst, USA. His current research interest includes the performance evaluation of communication networks. He is a member of IPSJ, and ACM SIGCOMM.