In smart home, location estimation based on PIR sensors is very popular. Existing methods by various sensors technologies and intelligent algorithms are used to achieve a high accuracy. In fact, how to deploy the PIR sensor is directly related to the accuracy. In this paper, we present an approach to deploying the PIR sensor based on the accessible priority by genetic algorithm. This paper presents a genetic algorithm that searches for an optimal or near optimal solution to the PIR sensor deployment for smart home. The fitness function of GA is based on the accessible priority of indoor areas. The accessible priority value of different area is set according to the indoor environment and daily accessible habits. The performance of the genetic algorithm was evaluated using several metrics, and the simulation results demonstrated that the proposed algorithm can optimize the network coverage in terms of accessible frequency.

1. Introduction

According to the United Nations online database [1], currently the percentage of elderly population is 7.6% that is projected to rise by 16.2% in 2050. Issues such as increased healthcare expenditure, burden to caregivers, and insufficient and inefficient care are more likely to occur as the health of older persons normally deteriorates with increasing age, resulting in more demand for long-term care. Therefore, it becomes urgent to find a compromise between the need for constant care (at home or in institutions) and the need to lighten the load on caregivers. Also, it is highly important to ensure that the ministered care is personalized and efficient. In this regard, the smart home concept is an adequate solution. It can be especially useful for the elderly or the disabled who wish to live independently.

In smart home, the occupant's location is probably one of the most important data items needed to monitor the occupant's activities [2]. Some home devices, such as lighting, heating, or home electronic devices, should be also adjusted automatically according to occupant's indoor position. So, it is necessary to localize people in smart home by sensor technologies and some intelligent methods. When the sensors are used for indoor localization, some techniques and constraints need to be considered, such as the occupant's privacy [3], hardware installment cost [4], and technologies' unobtrusiveness [5]. For PIRs are costly, have no privacy problem, and also have very low power consumption, they are very useful to track targets in indoor environment. In Gopinathan et al's paper, four PIR detectors are used to develop a system able to detect the path of a single person moving in an area [6], while in Shankar et al's paper a low cost sensor cluster is used to extract velocity as well as the path of a single person [7]. Also an array of PIR is used to detect the number of people moving through a gate [8]. In order to improve the tracking accuracy, some efforts are placed on position calculation by data fusion algorithms. Kim et al. developed a Bayesian classifier to enhance the accuracy of PIR-based indoor location-aware system [9]. Seifzadeh et al. proposed a Soft-Data-Constrained Multimodel Particle Filtering method, where the inherently vague (subjective) soft data provided by human agents are modeled using a fuzzy inference system [10].
These mentioned Bayesian methods offer a formal way to update existing knowledge from new evidence. Slightly finer localization can be obtained by exploiting the deployment of neighboring sensors. However, there are a limited number of studies that focus on coverage efficiency of PIR sensors. In this paper, we propose an implementation of GA-based approach for the PIR sensors deployment based on the priority of zones by accessible frequency in a smart home. We split the indoor environment in 2D view into small grids, and we assign a value to every grid that indicates the priority (accessible frequency) of the zone specified by that grid. For example, in a smart home there are many areas with different accessible frequencies. We can distinguish these areas based on obstacles (walls), furniture, the unoccupied free place, accessible daily habits, and so forth. The performance of a given PIR sensor layout by genetic algorithm was evaluated according to the area accessibility as the standard. Some simulation experiments have been implemented to evaluate the performance of our method for PIR deployments.

The rest of the paper is organized as follows: Section 2 will review related sensor deployment works. Section 3 will provide the description of the genetic algorithm with steps. Section 4 will discuss the experimental results and some simulation regarding the algorithm. Finally, Section 5 concludes this paper.

2. Related Works

This paper focused on the optimal PIR sensor placement with the accessible priority using GA-based approach for indoor localization. Therefore, the review of the related work consists of two parts: PIR sensor placement for indoor localization and GA for sensor placement.

2.1. PIR Sensor Placement for Indoor Localization. Generally, the PIR sensor placement is classified into two categories: (1) the deployment of PIR sensors: selecting the optimal coordinate for the PIR sensors and (2) the modulation of FOVs: modulating the optimal FOVs. In [11], the authors have mentioned that the deployment strategy of directional sensors nodes is critical to improve target detection and tracking accuracy. They presented the PIR sensor deployment in WSNs based on computational geometry. Byunghun and his workmates also show how the PIR sensors are deployed, which is very important for localizing an object using region-based tracking algorithm [12]. They discussed the PIR deployment that would divide the monitored room into different regions and how to deploy them determined the localization error. The work reported in [13] addresses the problem of target detection and classification using seismic and PIR sensors and makes use of wavelet-based feature extraction method to resolve this problem. The advantages and disadvantages of seismic and PIR sensors are also discussed. For example, the deployment of sensors largely depends on the intent of use and environment. PIR sensor deployment could be very expensive, because the detection range of PIR sensors has a short range of detection. Therefore, it is recommended that seismic and PIR sensor should be used together. In [14], PIR sensor nodes are deployed linearly and the sensing area is divided into cells through the four sensing regions of each PIR sensor node. In another study [15], five PIR sensors are attached to a service robot to make a circular sensing region. According to the experimental results stated in these studies, the usage of PIR sensors can improve the accuracy of target detection and tracking. Unlike real sensor nodes, most of the researches assume that sensor nodes have a 360° Angle of View (AoV). For example, a node deployment strategy that assumes PIR sensor nodes have 360° AoV is proposed in [7]. The performance of the proposed deployment algorithm is evaluated by comparing the coverage at the initial and final locations of sensor nodes.

2.2. GA for Sensor Placement. Sensor placement for indoor localization is an optimization problem that is nondeterministic and polynomial-time hard [16]. Genetic algorithms (GAs) are a type of evolutionary, optimization algorithm, which can be used to solve optimization problems with NP-hard complexity. Some previously developed algorithms have used GAs to obtain optimum deployment of sensor networks. The study in [17] presents a multiobjective genetic algorithm (MOGA) for optimal deployment of N static sensors in a 2D flat RoI, with two competing deployment objectives: maximizing the area coverage and maximizing lifetime. Each deployment layout was ranked according to its area coverage and lifetime. Different optimum layouts are obtained with respect to different sensing-to-communication range ratios. Reference [18] proposes a dynamic sensor nodes selection strategy, which uses the GA to implement global searching space of GA and ensure the validity of each chromosome. The two objectives of their fitness function are to maximize the coverage and minimize the area covered by more than one sensor. Liang et al. [19] have performed the optimization work for wireless sensor networks in 3D environments. In this work, a force driven genetic algorithm was introduced to decide the location of the sensors in order to obtain an optimized accessible space and also consider sensor coverage. Feng et al. propose an implementation of GA-based design approach for the deployment of the sensors and angles for the modulation of FOVs [20].

3. An Approach for Deploying PIR Sensor Using Genetic Algorithm

This section describes the approach that we propose to deal with the deployment issue of PIR-based positioning system. Two subsections are included in this part. Section 3.1 describes how the proposed approach is used to build the accessible map based on the human visiting habits. And Section 3.2 illustrates how our approach is used to optimally deploy the PIR sensors with the accessible priority of each area in indoor environment.

3.1. Building Accessible Map by Human Daily Habits. Given a mock apartment for deploying, our approach for building an accessible map is executed by human daily habits by performing the following steps, as illustrated in Figure 1.
Step 1. The full description of space topography of the smart home to be monitored is provided, shown as in Figure 1(a). This includes shape and size of the room layout, location of walls and furniture, and size of the areas in which the indoor environment is partitioned.

Step 2. The graphical characterization of the space topography is transformed into an accessible map, which can be used to resolve the accessible priority of each area in the smart home.

Step 3. The accessible map is provided as input of the GA so as to find as output the accessible priority in each area of the smart home. The image of the accessible map is resolved as follows.

Step 3.1. The space topography is fixed for a given indoor environment. If some places are occupied by the walls or furniture such as beds, chairs, or refrigerators, people do not or rarely visit these positions. So the areas of the accessible map occupied by walls or furniture are set as “0.” For other areas without the furniture and other obstacles, the initial value of the accessible map is set as “0.5.” The results are shown as in Figure 1(b).

Step 3.2. For the accessible areas in the smart home, not all positions have the same opportunities to be visited. We use another source of information, that is, the daily visiting habits of human subject when he interacts with the environment. Generally, the human subject visits more frequently around certain furniture. The mock apartment we designed has some key furniture: bedroom bed, living room couch, dining room chair, living room table, kitchen basin, and bathroom sink. The areas around the furniture have more visiting frequencies than other inner areas in the indoor environment.

Step 3.3. Each two pieces of furniture can generate arbitrary route. Some daily routes are trained by A* algorithm [21].
The starting and ending landmarks can be (p1) entrance to kitchen, (p2) kitchen to living room table, (p3) major bedroom to guest room, (p4) living room sofa to bathroom, and (p5) guest room to balcony, shown as in Figure 1(c). Tests with different trajectories were processed. The areas in these trajectories have more accessible frequencies than other areas of the apartment. These values can be obtained by observation.

Step 3.4. The last consideration is for transitions between the accessible areas and their surrounding areas. The accessible map value can be computed by convolution of the accessibility with a Gaussian kernel in two dimensions. The mean of the Gaussian kernel is the related wall’s position minus/plus 0.2 m, and the variance of the Gaussian kernel is 0.2 m. The equation can be described by

$$A_{map}(x, y) = \sum_{k=x-20}^{x+20} \sum_{l=y+20}^{y-20} h(x - k, y - l) \cdot A_{map}(k, l).$$

Step 4. Based on the prior knowledge, the accessible map can be represented in Figure 1(d). The accessible map is discretized by the assumed occupant visits respective grid points. Each grid size is $\Delta x \times \Delta y$, and $\Delta x = \Delta y = 0.2$ m. The digital binary image of the heat “map” is shown as in Figure 1(d), denoted by the matrix $B$. Each element $B(x_i, y_i)$ denotes the “heat” score of the $i$th grid point for deployed space if

$$B(x_i, y_i) > 0.$$ (2)

That means the $i$th grid point can be accessible. Then we generate the array Code to save the coordinate

$$Code(i) = [x_i, y_i],$$ (3)

where $i \in [1, N_{code}]$, in which $N_{code}$ is denoted by the accessible number of the deployed space. These array values will be used in the formulation of the optimization problem.

3.2. PIR Sensors Optimal Deployment Based GA-Based Algorithm. Genetic algorithm (GA) is a stochastic search technique that mimics the natural evolution proposed by Charles Darwin in 1858 [22]. GA has been successfully applied to a wide range of combination problems. It is modeled largely on the principles of the evolution via natural selection [23], employing a population of individuals that carry on the selection procedure in terms of variation-inducing operators such as mutation and recombination (crossover). A fitness function is used to evaluate individuals, and reproductive success varies with fitness. The main objective of employing the GA in our approach is to enhance the performance of the indoor location after initial deployment of the stationary nodes. The details are described as follows.

3.2.1. PIR Sensor Parameters. The interested parameters of the PIR sensing model are represented with the use of a vector $p$:

$$p_i = [x_i, y_i, z_i, \theta_i]^T,$$ (4)

where the $i$th PIR sensor is placed on $(x_i, y_i, z_i)$ and the FOV of $i$th PIR sensor is denoted by $\theta_i$. The PIR sensors are all fixed on the same ceiling and therefore omit the $z$-coordinate. If the $z$-coordinate is fixed, the FOV of the PIR sensor can be denoted by the radius $R$ of sensing circle area. The PIR sensing area in our setup is modeled as a circle on the floor with a radius of 1 m. So the parameters of PIR deployment can be simply represented as

$$p_i = [x_i, y_i]^T.$$ (5)

3.2.2. Chromosomes Modeling. Each chromosome, as a solution in the GA, represents a set of deployment locations of potential PIRs in the specific apartment. The gens of each chromosome are positive integer values $g_i$ that correspond to the location $(x_i, y_i)$ of the $i$th PIR sensor on the $x$-$y$ plane. For our specific mock apartment, we encode the grid points of the space in 2D view. The maximum of code is $M$. Each chromosome in our approach can be denoted by

$$C = [g_1, \ldots, g_L]$$ (6)

where $L$ denotes the number of PIR sensors deployed and the gen $g_i$ denotes each PIR location code in the deployed plane, where $1 \leq g_i \leq M$:

$$g_i \rightarrow (x_i, y_i).$$ (7)

For example, the grid of $g_i = 11$ represents PIR node mapped to the location $(3, 3)$ in $x$-$y$ plane; the corresponding chromosome is shown in Figure 2.

3.2.3. Fitness Function. The fitness function is used in order to choose the best fittest chromosomes for the purpose of reproduction of the next generated solutions by the GA. In our approach, the fitness function calculates the sum of “heat” scores for each grid point in the $x$-$y$ plane of the specific space. The “heat” score $h_i$ equals the visiting frequency of each grid from the accessible map. Then the sum of all grids heat scores is calculated by decreasing plus strategy, which means the value of heat score is decreased by 1 if the coverage level increased and one limitation requirement is that the coverage level number is not greater than the hot rank of the position. Then this fitness function is given by

$$G = \sum_{i=1}^{N_{heat}} \sum_{j} (H_{rank}(i) + 1 - j)^2,$$ (8)

where $N_{heat}(i)$ denotes the number of the accessible grid points in the accessible map, $N_{\text{cover}}(i)$ is the number of PIRs
which covered the \( i \)th grid point, and \( H_{\text{rank}}(i) \) denotes the heat score rank of the \( i \)th grid point.

According to randomness and parallelism of evolution individual for PIR layouts, the overlapping probability of multiple PIRs sensing area is large. For this reason, the restriction condition for the number \( N_{\text{num}}(i) \) of overlapping PIR of the \( i \)th grid point is set. That means if \( N_{\text{num}}(i) \) is greater than the maximum of threshold \( N_{\text{max}} \), some punishments have been made. So the fitness function can be modified as

\[
G = \begin{cases} 
\sum_{i=1}^{N_{\text{heat}}} \sum_{j=1}^{N_{\text{cover}}(i)} (H_{\text{rank}}(i) + 1 - j)^2, & N_{\text{cover}}(i) \leq H_{\text{rank}}(i), N_{\text{cover}}(i) \leq \text{over}, \\
\sum_{i=1}^{N_{\text{heat}}} \left( \sum_{j=1}^{H_{\text{rank}}(i)} (H_{\text{rank}}(i) + 1 - j)^2 - \sum_{r=1}^{N_{\text{cover}}(i)} k \cdot H_{\text{rank}}^2(r) \right), & H_{\text{rank}}(i) \leq \text{over}, N_{\text{cover}}(i) > H_{\text{rank}}(i), \\
\sum_{i=1}^{N_{\text{heat}}} \left( \sum_{j=1}^{\text{over}} (H_{\text{rank}}(i) + 1 - j)^2 - \sum_{r=1}^{N_{\text{cover}}(i)} k \cdot H_{\text{rank}}^2(r) \right), & H_{\text{rank}}(i) > \text{over}, N_{\text{cover}}(i) > \text{over}. 
\end{cases}
\]

In this way, the problem of the optimal sensor deployment is formulated as

\[
\text{maximize } G. \quad (10)
\]

In order to avoid the prematurity of the population evolution phenomenon and keep the diversity of population, the fitness function is processed by the power function compression. Then the final expression of the fitness function is

\[
f = f^\varphi_{\text{initial}}. \quad (11)
\]

\( \varphi \) denotes the compression scale

\[
f_{\text{initial}} = \begin{cases} 
G, & G > 0, \\
0.0001, & G \leq 0.
\end{cases} \quad (12)
\]

3.2.4. GA Selection. We use Roulette Wheel Selection (RMS) as the selection method, which gives to each individual \( i \) of the current population a probability of being selected proportional to its fitness \( f(i) \):

\[
p(i) = \frac{f(i)}{\sum_{j=1}^{n} f(j)}, \quad (13)
\]

where \( n \) denotes the population size in terms of the number of individuals. We used the fitness ratio calculated in (13) as a measure for ranking the chromosomes and then performing parent selection according to the ratio participated by each chromosome in the fitness function. The chromosomes with higher ranking have a higher probability to become parents and perform reproduction of new individuals than others with smaller ranking. This property must be maintained in order to increase the improvement ratio at each generation of the GA.
3.2.5. GA Operator Operations

(a) Crossover. The crossover operator is a genetic operator that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). Arithmetic crossover (AC) operator is used for GA in our approach. Children are produced a new chromosome (offspring). Arithmetic crossover that combines (mates) two chromosomes (parents) to produce a new chromosome (offspring). Children are produced according to a user-definable crossover probability \( p_c \). \( p_c \) is increased when the population tends to get stuck at a local optimum and decreased when the population is scattered in the solution space. Here \( p_c \) is defined as follows:

\[
P_c = \begin{cases} 
\frac{k_1 \sin \left( \frac{\pi}{2} \cdot \frac{f_{\text{max}} - f}{f_{\text{max}} - f_{\text{avg}}} \right)}{k_2}, & f' > f_{\text{avg}} \\
\frac{p_{c1} \sin \left( \frac{\pi}{2} \cdot \frac{f_{\text{max}} - f}{f_{\text{max}} - f_{\text{avg}}} \right)}{p_{c2}}, & f' \leq f_{\text{avg}} 
\end{cases}
\]

where \( f_{\text{avg}} \) is an average fitness value of the population, \( f_{\text{max}} \) is maximum fitness value of the population, \( f' \) is the largest of the fitness values of the solutions to be crossed, \( 0 < k_1, k_2 < 1 \), and \( k_2 \) is the default crossover probability. In order to make the optimal individual have higher crossover probability, \( p_c \) is refined by

\[
P_c = \begin{cases} 
\frac{p_{c1} \sin \left( \frac{\pi}{2} \cdot \frac{f_{\text{max}} - f}{f_{\text{max}} - f_{\text{avg}}} \right)}{p_{c2}}, & f' > f_{\text{avg}} \\
\frac{p_{c1}}{p_{c2}}, & f' \leq f_{\text{avg}} 
\end{cases}
\]

A threshold related with the current reproduction generation is

\[
\text{th} = \frac{M_{\text{gen}}}{20} \cdot \sin \left( \frac{\pi}{2} \cdot \frac{\text{gen}}{M_{\text{gen}}} \right),
\]

where \( M_{\text{gen}} \) is the maximum value of reproduction generations, \( \text{gen} \) is the current value of reproduction generations, and \( M_{\text{gen}}/20 \) is an adaptive coefficient. We also define a variable \( \text{repnum} \), which represents the holding time (reproduction generation) of the optimal individual of population. The last adaptive step for \( p_c \) is

\[
P_c = \begin{cases} 
p_{c1}, & \text{repnum} \leq \text{th}, \\
p_{c2}, & \text{repnum} > \text{th} \end{cases}
\]

(b) Mutation. Mutation introduces random binary changes in a chromosome, which helps escape from local minima’s trap and maintains diversity in the population. For binary representation, a simple mutation can consist in inverting the value of each gene with a small probability. In this paper, an improved mechanism of mutation is introduced. Move the actual coordinates of randomly selected individuals representing a real gene; displacement amplitude is based on the individual fitness of scale. The larger the fitness of the individual chromosome is, the smaller the displacement amplitude would be. The probability of mutation is adjusted by using the following equation:

\[
P_m = p_{m1} \sin \left( \frac{\pi}{2} \cdot \frac{f_{\text{max}} - f}{f_{\text{max}} - f_{\text{min}}} \right),
\]

where \( f_{\text{min}} \) is the minimum value of the chromosome fitness and \( p_{m1} \) is mutation probability constant. \( p_m \) is only related to the displacement amplitude; the relationship is based on the following expression:

\[
\text{step} = \text{round} \left( b \cdot p_m \cdot l \right) + 1,
\]

where \( \text{step} \) denotes the displacement amplitude, \( b \) is random, and \( l \) represent displacement length:

\[
P_m = \begin{cases} 
p_{m1}, & \text{repnum} \leq \text{th}, \\
p_{m2}, & \text{repnum} > \text{th} \end{cases}
\]

4. Simulation Results

The objective of this work is to have an optimal deployment for PIR-based indoor localization. We consider in the simulation 12 m \( \times \) 7.2 m area. The thickness of wall is 0.2 m and the door of all the space, such as the kitchen and bathroom, is 0.2 \( \times \) 1.0 m. By calculating exact size of various places, all the possible coverage area for PIR sensors is 75.12 m². The PIR sensing area in this setup is a shape of circle on the floor with a radius of 1 m. Table 1 summarizes the parameters in the implementation of GA. These parameters were obtained empirically after several experiences.

In order to analyze the performance of our method, several simulation experiments were conducted for the performance evaluation. Hence, we compared the simulation running time and fitness function value when the deployed PIR sensors’ number is different for the special indoor environment first. Then we investigated how the accessible heatmap rank affects the PIR deployment in case of a given scenario. Next, we took a closer look at how to keep the tradeoff between the coverage and overlap of the PIR sensing area. And finally, we compared the performance of three deployment methods.
Therefore, the experiment was done to get an appropriate decrease will lead to the increase of the PIR overlap range. Because our rule is that only the low rate zones’ overlap corresponding heat division would be accurate when the “heat” score of the accessible map, a set of values related to the accessible frequency of the areas. According to the “heat” score range of the accessible map, a set of values 

\[ r_1, \ldots, r_S \]

is defined. \( S \) is the scale size of the “heat” score range. The value of \( r_1 \) increases as the “heat” score of the grid point is larger. Figure 5(a) shows the accessible map at rank \( S = 4 \), and Figure 5(b) describes the corresponding PIR sensor deployment.

Table 1: Setting for GA parameters.

| Population size | 300 individuals |
|-----------------|-----------------|
| Generations     | 400             |
| Individual chromosome length | 2–10 |
| Crossover probability | \( P_{c1} = 0.1 \) \( P_{c2} = 0.9 \) |
| Mutation probability | \( P_{m} = 0.5 \) |
| PIR sensor radius | 1 meter |
| Overlapping penalty coefficient | \( k = 7 \) |
| Heat layered ranking | 4 |
| Coverage layer constraint | 2 |

4.1. The Number of PIRs for Optimal Deployment. The first experiment analyzed the effect of PIR sensors’ number. As expected, the PIR sensors’ number increased the algorithm run time proportionally, listed as in Table 2. In case of a given indoor environment, when the PIR sensors’ number is lower than 7, the increase in the number of deployed PIR sensors will not affect the deployment of PIR sensors. However, the number of PIR sensors is larger than 7 and the deployment of PIR sensor will vary with the PIR sensor numbers. In our lab, the PIR sensors’ number needs to be determined according to the fitness function.

In Figure 4, the fitness value changes for different population size and PIR sensors’ number. As is shown, a specific plot represents the fitness versus time under the fixed PIR sensor number. There are three sections in one curve. Section I shows the reproduction in region A, Section II shows the reproduction in region B, and Section III represents the results of mutation with region A and region B. The fitness increases as the PIR sensor number increases.

4.2. Effect of Rank of the Accessible Heatmap. The second experiment was performed to understand the effect of the accessible heatmap score rank. The heat score \( S \) is used to equally divide the original heat zones based on the different degrees of heat. With the heat range being fixed, the corresponding heat division would be accurate when \( S \) increases. This can effectively eliminate the multiple priority layouts. Because our rule is that only the low rate zones’ overlap should be punished, the relatively low rate heat range’s decrease will lead to the increase of the PIR overlap range. Therefore, the experiment was done to get an appropriate \( S \). The number of PIR sensors is set as 10 in the following experiments. \( r_j \) denotes the rank of “heat” score of the \( j \)-th grid point, related to the accessible frequency of the areas. According to the “heat” score range of the accessible map, a set of values \( \{r_1, \ldots, r_S\} \) is defined. \( S \) is the scale size of the “heat” score range. The value of \( r_j \) increases as the “heat” score of the grid point is larger. Figure 5(a) shows the accessible map at rank \( S = 4 \), and Figure 5(b) describes the corresponding PIR sensor deployment.

Figure 5 directly presents the features of the PIR layout when \( S \) is set as different numbers. When more accurate heat stratification is needed to eliminate the multiple priority, one method is to increase the rank \( S \). Figures 5(c) and 5(e) show that the heat stratification is growing accurate with the increase of the rank \( S \). However, increasing the rank will make the PIR layout tend to overlap. Compared to Figures 5(b) and 5(d), the appropriate overlap theory can enhance the localization accuracy. But this increasing to a certain degree, serious overlap would occur, shown as in Figure 5(f). At this point, it is the same as accuracy. After comparison, the heatmap rank degree is set as 6.

4.3. PIR Coverage and Overlap Rate by Different PIR Number. The third experiment examined the effect of the PIR sensors’ number against the sensing coverage and overlap. Three parameters are defined to evaluate the effect of PIR sensor number on the sensor coverage. They are listed as follows:

(a) Maximum coverage ratio \( R_{cov,max} \) is the ratio of area that can be covered by all PIRs without overlap \( A_{Covered} \) over the entire sensing field \( A_{Total} \):

\[
R_{cov,max} = \frac{A_{Covered}}{A_{Total}} \tag{22}
\]

(b) Actual coverage rate \( c_{act} \), which is defined as sum of PIRs sensing areas based on the solution deployment \( c_{sp} \) over the deployment area of the specific space \( s \):

\[
c_{act} = \frac{c_{sp}}{s} \tag{23}
\]

(c) Overlap rate \( o_c \), which is defined as the sum of PIRs overlap areas \( o_c \) over sum of PIRs sensing areas based on the solution deployment \( c_{sp} \):

\[
o_c = \frac{o_c}{c_{sp}} \tag{24}
\]

The results can be seen in Table 3. The value \( o_c \) is not too large when the PIRs number is less than 8. The results show that the rule of increasing PIR coverage has more priority; the proposed algorithm improves the coverage of PIRs to
Table 2: PIR sensors’ deployment versus the PIR sensors’ number.

| PIR number | PIR1     | PIR2     | PIR3     | PIR4     | PIR5     | PIR6     | PIR7     | PIR8     | PIR9     | PIR10     | Best fit | Running time |
|------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|----------|--------------|
| 2          | (15, 27) | (15, 36) |          |          |          |          |          |          |          |           | 8.9111   | 3 min        |
| 3          | (15, 27) | (15, 36) | (18, 11) |          |          |          |          |          |          |           | 9.9193   | 4 min        |
| 4          | (15, 27) | (15, 36) | (18, 11) | (20, 49) |          |          |          |          |          |           | 10.6935  | 5 min        |
| 5          | (15, 27) | (15, 36) | (18, 11) | (20, 49) | (30, 24) |          |          |          |          |           | 11.3356  | 6 min        |
| 6          | (15, 27) | (15, 36) | (18, 11) | (20, 49) | (30, 24) | (27, 37) |          |          |          |           | 11.9006  | 7 min        |
| 7          | (15, 27) | (17, 36) | (18, 11) | (20, 49) | (30, 24) | (27, 37) | (10, 36) |          |          |           | 12.3653  | 8 min        |
| 8          | (15, 27) | (17, 36) | (18, 11) | (20, 49) | (30, 24) | (27, 37) | (10, 36) | (30, 13) |          |           | 12.7523  | 9 min        |
| 9          | (14, 28) | (17, 37) | (18, 11) | (20, 49) | (30, 24) | (27, 37) | (9, 36)  | (30, 13) | (20, 26) | (15, 54)  | 13.0892  | 10 min       |
| 10         | (14, 28) | (17, 37) | (18, 11) | (21, 48) | (27, 37) | (30, 24) | (9, 36)  | (19, 26) | (30, 13) | (15, 54)  | 13.3965  | 11 min       |

Figure 5: Quantization effect of the heatmap at different degree and corresponding PIR deployment.

enhance indoor localization. There is no change in overlap rate when the PIRs number is 8 to 10. However, if the number of PIRs is greater than 10, the overlap rate will increase. So the optimal number of PIRs for our simulation experiment is 10.

The data in Table 3 indicate that the overlap rate is small and there occurs no overrated overlap if PIRs number is lower than 8. This shows that the overlap rate tends to be low if the PIRs number is small, which goes in line with the coverage priority principle. That is, the coverage needs to be more if there are fewer PIRs. The overlap rate and overrated overlap remain unchangeable when the PIRs number is 8, 9, or 10. At this point, the result would be superb if the PIRs number is 10 based on the actual coverage choice standard. The actual coverage does not increase much and the overrated overlap is high when the PIRs number increases to 11. Compared to the outcome of the PIRs number being 10, the localization
Table 3: Evaluation parameters for PIR coverage and overlap rate.

| PIR number | c_{max} | c_{act} | oc  |
|------------|---------|---------|-----|
| 2          | 0.0836  | 0.0829  | 0.0094 |
| 3          | 0.1255  | 0.1247  | 0.0063 |
| 4          | 0.1673  | 0.1666  | 0.0043 |
| 5          | 0.2091  | 0.1960  | 0.0067 |
| 6          | 0.2509  | 0.2377  | 0.0556 |
| 7          | 0.2927  | 0.2795  | 0.0473 |
| 8          | 0.3346  | 0.3499  | 0.0673 |
| 9          | 0.3764  | 0.3567  | 0.0625 |
| 10         | 0.4182  | 0.3903  | 0.0552 |
| 11         | 0.4600  | 0.4151  | 0.1083 |

Figure 6: The fitness curves by GA, greedy algorithm, and manual random deployment.

accuracy has not much increased. Hence, we would believe the best PIRs number which is 10 would be the best layout.

4.4. Performance by Three Deployment Methods. In the last experiment, changes in the fitness by GA, greedy algorithm, and manual random deployment were observed. The performance curves by three various PIR deployment algorithms are presented in Figure 6. The green curve is with the GA method and the red and blue curves are represented by greedy algorithm and random deployment, respectively. Our purpose is to find a common way to solve the PIR sensor layout problem from simple to complex. When people do not know how to make PIR sensor layout, the simplest way is to put them in an optional space by random deployment. Then, the possible method is to place the sensor according to the fitness value in order which is named greedy algorithm. After careful consideration of the previous method, it can be found that all the sensors are not placed at the same time, so it is likely to lead to a local optimal state. GA is a parallel algorithm which can solve this problem easily. Although the performance between GA and greedy algorithm is similar at the beginning, the gap is increasing with the increase of the number of PIR sensors. In fact, the gap has a greater impact on the accuracy of the location experiment when the sensor layout is determined. The results show the PIR sensors layout is not ideal by manual random deployment and also not stable. As the PIR sensors number increases, the performance by our proposed method is better than the other two.

5. Conclusion

The main contribution of this paper was to optimize the coverage in sensor networks considering area accessible frequency. The GA we proposed to solve the coverage problem defined in this study stands out with its novel fitness function, which is quickly calculable and based on area priority. Additionally, the proposed optimizations are novel approaches that we introduced in this study that allow for faster convergence of the solutions. The actual coverage rate is 39.03%, and overlap rate is 7.41% when the number of PIR sensors is 10. The running time of our method for PIR sensor layout is greater than 10 minutes. It will increase according to the number of PIR sensors. In the future, we tried to obtain the PIR sensors layout by the multiobject function and some considerations will be placed on the efficacy of the algorithm.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] Population Division of the Department of Economic and Social Affairs of the United Nations Secretariat, *World Population Prospects: The 2008 Revision*, Population Division of the Department of Economic and Social Affairs of the United Nations Secretariat, 2008.
[2] Y. Rahal, H. Pigot, and P. Mabilleau, “Location estimation in a smart home: system implementation and evaluation using experimental data,” *International Journal of Telemedicine and Applications*, vol. 2008, Article ID 142803, 9 pages, 2008.
[3] S. Bahadori, L. Iocchi, G. R. Leone, D. Nardi, and L. Scozzafava, “Real-time people localization and tracking through fixed stereo vision,” *Applied Intelligence*, vol. 26, no. 2, pp. 83–97, 2007.
[4] Y. Kaddourah, J. King, and A. Helal, “Cost-precision tradeoffs in unencumbered floor-based indoor location tracking,” in *Proceedings of the 3rd International Conference on Smart Homes and Health Telematic (ICOST ’05)*, Sherbrooke, Canada, July 2005.
[5] T. Kivimäki, T. Vuorela, P. Peltola, and J. Vanhala, “A review on device-free passive indoor positioning methods,” *International Journal of Smart Home*, vol. 8, no. 1, pp. 71–94, 2014.
[6] U. Gopinathan, D. J. Brady, and N. P. Pitsianis, “Coded apertures for efficient pyroelectric motion tracking,” *Optics Express*, vol. 11, no. 18, pp. 2142–2152, 2003.
[7] M. Shankar, J. B. Burchett, Q. Hao, B. D. Guenther, and D. J. Brady, “Human-tracking systems using pyroelectric infrared
detectors,” *Optical Engineering*, vol. 45, no. 10, Article ID 106401, 2006.

[8] K. Hashimoto, K. Morinaka, N. Yoshiike, C. Kawaguchi, and S. Matsueda, “People count system using multi-sensing application,” in *Proceedings of the International Conference on Solid State Sensors and Actuators (TRANSUDCERS ’97)*, vol. 2, pp. 1291–1294, IEEE, Chicago, Ill, USA, June 1997.

[9] H. H. Kim, K. N. Ha, S. Lee, and K. C. Lee, “Resident location-recognition algorithm using a Bayesian classifier in the PIR sensor-based indoor location-aware system,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 39, no. 2, pp. 240–245, 2009.

[10] S. Seifzadeh, B. Khaleghi, and F. Karray, “Soft-data-constrained multi-model particle filter for agile target tracking,” in *Proceedings of the 16th International Conference of Information Fusion (FUSION ’13)*, pp. 564–571, Istanbul, Turkey, July 2013.

[11] S. Akbas, M. Akif Efe, and S. Ozdemir, “Performance evaluation of PIR sensor deployment in critical area surveillance networks,” in *Proceedings of the IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 327–332, Marina Del Rey, Calif, USA, May 2014.

[12] S. Byunghun, C. Haksoo, and S. L. Hyung, “Surveillance tracking system using passive infrared motion sensors in wireless sensor network,” in *Proceedings of the International Conference on Information Networking (ICOIN ’08)*, pp. 1–5, January 2008.

[13] X. Jin, S. Sarkar, A. Ray, S. Gupta, and T. Damarla, “Target detection and classification using seismic and PIR sensors,” *IEEE Sensors Journal*, vol. 12, no. 6, pp. 1709–1718, 2012.

[14] Z. Zhang, X. Gao, J. Biswas, and J. K. Wu, “Moving targets detection and localization in passive infrared sensor networks,” in *Proceedings of the 10th International Conference on Information Fusion*, pp. 1–6, IEEE, Québec, Canada, July 2007.

[15] A. Ş. Sekmen, M. Wilkes, and K. Kawamura, “An application of passive human–robot interaction: human tracking based on attention distraction,” *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 32, no. 2, pp. 248–259, 2002.

[16] M. Younis and K. Akkaya, “Strategies and techniques for node placement in wireless sensor networks: a survey,” *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.

[17] D. B. Jourdan and O. L. de Weck, “Layout optimization for a wireless sensor network using a multi-objective genetic algorithm,” in *Proceedings of the 59th IEEE Vehicular Technology Conference (VTC-Spring ’04)*, vol. 5, pp. 2466–2470, IEEE, Milan, Italy, May 2004.

[18] X. Wang, S. Wang, and D. Bi, “Dynamic sensor nodes selection strategy for wireless sensor networks,” in *Proceedings of the International Symposium on Communications and Information Technologies*, pp. 1137–1142, October 2007.

[19] S. Liang, A. Nur Zincir-Heywood, and M. I. Heywood, “Intelligent packets for dynamic network routing using distributed genetic algorithm,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO ’02)*, pp. 88–96, July 2002.

[20] G. Feng, M. Liu, and G. Wang, “Genetic algorithm based optimal placement of PIR sensors for human motion localization,” *Optimization and Engineering*, vol. 15, no. 3, pp. 643–656, 2014.

[21] http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html.

[22] https://en.wikipedia.org/wiki/Charles_Darwin.

[23] M. Mitchell and S. Forrest, “Genetic algorithms and artificial life,” *Artificial Life*, vol. 1, no. 3, pp. 267–289, 1994.