

Research Article

Green Clustering Implementation Based on DPS-MOPSO

Yang Lu,¹ Xuezhi Tan,^{1,2} Yun Mo,¹ and Lin Ma^{1,2}

¹ Communication Research Center, Harbin Institute of Technology, Harbin 150080, China

² Key Laboratory of Police Wireless Digital Communication, Ministry of Public Security, Harbin 150080, China

Correspondence should be addressed to Xuezhi Tan; tanxz1957@hit.edu.cn

Received 12 May 2014; Revised 11 August 2014; Accepted 7 September 2014; Published 21 October 2014

Academic Editor: Gisele Mophou

Copyright © 2014 Yang Lu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A green clustering implementation is proposed to be as the first method in the framework of an energy-efficient strategy for centralized enterprise high-density WLANs. Traditionally, to maintain the network coverage, all of the APs within the WLAN have to be powered on. Nevertheless, the new algorithm can power off a large proportion of APs while the coverage is maintained as the always-on counterpart. The proposed algorithm is composed of two parallel and concurrent procedures, which are the faster procedure based on K -means and the more accurate procedure based on Dynamic Population Size Multiple Objective Particle Swarm Optimization (DPS-MOPSO). To implement green clustering efficiently and accurately, dynamic population size and mutational operators are introduced as complements for the classical MOPSO. In addition to the function of AP selection, the new green clustering algorithm has another new function as the reference and guidance for AP deployment. This paper also presents simulations in scenarios modeled with ray-tracing method and FDTD technique, and the results show that about 67% up to 90% of energy consumption can be saved while the original network coverage is maintained during periods when few users are online or when the traffic load is low.

1. Introduction

Increasing numbers of enterprise offices [1], university campuses (Dartmouth College WLAN), and municipal downtowns [2] deploy WLANs for flexible Internet connectivity. Specifically, in some offices and campuses, WLANs are deployed consisting of hundreds and even thousands of APs (Access Points) which make the WLAN high-density to meet the increasing demand of network accesses. Many of those WLANs, so-called traditional WLANs, are designed to improve the usage of users during times of peak demand, which rarely occurs [3]. All APs in traditional WLANs are always on to ensure the coverage of the network, and many of the APs are considered idle or redundant when no users are in connection with them. The utilization rate is even lower during weekends or school holidays, representing an avoidable type of energy wastage.

Concerning energy efficiency in WLANs, D'Alessandro et al. introduce a multicarrier infrastructure to implement energy conservation in WLANs [4]. According to the analysis in [5], Miliotis et al. study the relations between communication efficiency and energy consumption which are important

to wireless network management in wireless communication. In [6], Rabaey considers that most energy consumption of wireless communication device is in standby mode and reducing the energy consumption of components is an important way to implement green communication. Marsan et al. introduce an analytical model for traffic in WLANs and present the threshold determined by the number of clients and traffic of an AP switch algorithm [7]. Chen et al. [8] and Kushki et al. [9] use information theory and probability theory to implement AP selection and Kushki et al. quantify the resolution of location on the basis of Fisher Rule for AP selection [10]. In [11], Fang et al. use a prediction-based method to cluster APs in a decorrelated space.

As an alternative utilization, the energy-efficient strategy can be applied in high-density WLANs for power conservation. The main components of such a strategy usually include procedures, which are initialization and preparation, the access and association management of end users, the analysis mechanism of network information, and the topology management of APs. A green clustering algorithm is introduced and performs the function as the part of initialization, which is mainly responsible for forming AP clusters and selecting

the cluster-head APs. Green clustering is the first algorithm for energy-efficient strategies and a topology that consists of cluster-head APs and can maintain the coverage of network the same (almost the same) as the traditional WLANs can be formed.

In this paper, a new green clustering algorithm based on DPS-MOPSO (Dynamic Population Size Particle Swarm Optimization) is proposed, which will be referred to as GC-DPS-MOPSO. It consists of two different procedures, which are the faster procedure implemented through K -means algorithm and the more accurate procedure implemented based on DPS-MOPSO. The new approach forms AP clusters and selects cluster-head AP in a completely new perspective. Actually, either of the two procedures may achieve the designed requirements of the new algorithm, but reasonably combining the two approaches and integrating as the proposed GC-DPS-MOPSO can lead to a further improvement in terms of both efficiency and accuracy.

The remainder of this paper is arranged as follows. An overview of related work and background knowledge are given in Section 2. Classical PSO and MOPSO are introduced, and DPS-MOPSO is proposed in Section 3. In Section 4 we propose GC-DPS-MOPSO based on K -means algorithm and DPS-MOPSO. Section 5 provides the simulation results and performance analysis. Section 6 presents the conclusions and discussion of the paper.

2. Background and Related Works

In many offices and campuses WLANs are deployed consisting of hundreds or thousands of APs, which make the WIFI network high-density in terms of AP, and the scheme of AP deployment is designed to guarantee peak demand, which is the reason that traditional WLANs mostly cause serious AP redundancy. We propose an energy-efficient strategy in high-density WLANs to reduce the policy-based power consumption, and the green clustering based on DPS-MOPSO (GC-DPS-MOPSO) is an important and initial part of our strategy.

SEAR (Survey, Evaluate, Adapt, and Repeat) [12] is introduced firstly as the general foundation model to show the concept, definition, and working mechanism of an energy-efficient strategy. It was proposed by Jardosh as a demand-driven strategy to manage APs in high-density WLANs efficiently. SEAR is a policy-based method, which means it can be tailored to achieve the performance according to the requirements of WLAN administrators. Based on some certain policies used, SEAR can conserve energy while maintaining the same performance that clients achieve in its always-on counterpart. SEAR is assumed to have complete knowledge of the physics position and state of all APs managing the WLAN to achieve the purpose efficiently and switch the APs on or off with respect to the policy used. SEAR consists of four components, which are green clustering algorithm, user demand estimation, topology management, and user association, as shown in Figure 1. A good introductory survey for SEAR can be found in [12].

In our energy-efficient strategy, as the same as in SEAR, we propose a green clustering algorithm. Firstly, the concept

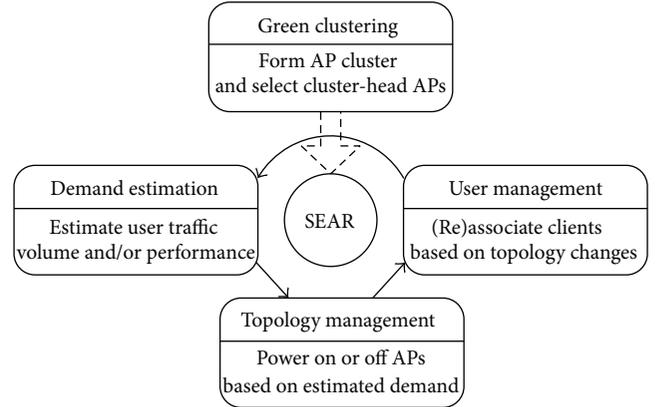


FIGURE 1: Components of SEAR.

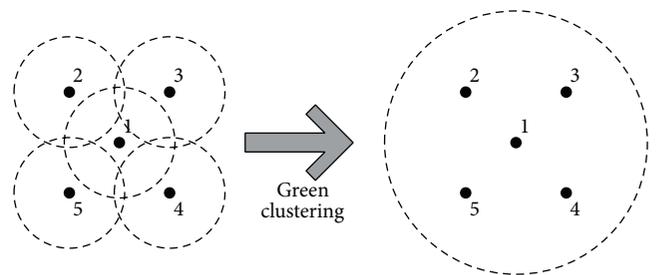


FIGURE 2: Illustration of cluster formation.

of green clustering is restated and clarified. The strategy-based energy efficiency is implemented through powering off redundant APs when few users are online or when the traffic is low. The green clustering is the algorithm to form AP clusters and determine a cluster-head AP for each green cluster. According to the results of green clustering, the cluster-head APs are powered on, whereas other APs (called secondary APs) are off for energy efficiency while maintaining the same coverage as shown in Figure 2. Besides, it is worth mentioning that the transmitting power of a net device is only a tiny part of the total power consumption. Taking Cisco WAP4410N Wireless-N AP of S series, for example, its transmitting power with a single antenna is 13 dBm to 17 dBm (12.6 to 50.1 in mW), while this AP's total power consumption is about 10 W during normal work. Therefore, even when the transmitting power increases for satisfying the algorithm requirement, the impact on total power consumption could be ignored because the energy savings by powering off APs are much more significant.

In addition, we verified and implemented our green clustering algorithm based on an Evolutionary Algorithm (EA) in our previous work [13], which is also considered as a comparison in this paper.

3. Proposed DPS-MOPSO

To implement green clustering, a method, known as DPS-MOPSO (Dynamic Population Size Multiple Objective Particle Swarm Optimization), is introduced. DPS-MOPSO is an

approach of multiobjective optimization, in which we make improvement based on the classical MOPSO with dynamic population size controlling and mutational operation. Firstly, the classical PSO is introduced as follows.

3.1. Classical PSO. PSO (Particle Swarm Optimization) is an evolutionary computation technique developed by Kennedy and Eberhart in 1995 and is inspired by the social behavior of bird flocking and fish schooling [14]. The algorithm of classical PSO is as the following steps.

- (1) Initialize the population POP , and $POP[i]$ can be obtained, where $i = 1$ to MAX and MAX denotes the total number of particles.
- (2) Initialize the speed of each particle, and $V[i]$ ($i = 1$ to MAX) can be initialized as well.
- (3) Evaluate each of the particles in POP .

Each particle is evaluated through assigning a fitness value, the value of which is equaled to the objective function value.

- (4) Store the particles in the repository REP .
- (5) **WHILE** maximum number of iterations has not been reached, **DO** the following.

- (A) Update the velocity of each particle using the following expression,

$$V[i] = wV[i] + c_1 rand_1(REP[i] - POP[i]) + c_2 rand_2(REP[h] - POP[i]), \quad (1)$$

where w is the inertia weight factor, c_1 and c_2 are constants known as acceleration coefficients, $rand_1$ and $rand_2$ are two separately generated uniformly distributed random numbers in the range of $[0, 1]$, $REP[i]$ denotes the best particle i has had, and $REP[h]$ represents the best particle in POP . The first part of (1), known as “inertia,” “momentum,” or “habit” denotes the previous velocity. The second part, known as the “cognitive” or “memory” component, demonstrates the personal thinking of each individual. The last part is known as the “social knowledge” or “cooperation” component, which represents the collaborative effect amongst particles.

- (B) Update $POP[i]$, using following expression:

$$POP[i] = POP[i] + V[i]. \quad (2)$$

- (C) Maintain the particles within the search space.
 - (D) Evaluate each of the particles in POP .
 - (E) Update the repository REP . **IF** $POP[i]$ is better than $REP[i]$, **THEN** $REP[i] = POP[i]$.
 - (F) Increment the loop counter.
- (6) **END WHILE.**

3.2. DPS-PSO. In the above subsection, the classical PSO iterates to search for an optimal or near-optimal solution. However, one of the main drawbacks is the excessive repetition during the update process of every particle within each iteration. Additionally, it is typical that, at the beginning of a classical PSO, most of the particles will be distributed far away from the potential global optimum. Then, as iterating, the particles will migrate to a region surrounding the global optimum. Therefore, later iterations serve and fine-tune the approach until the global optimum is reached. To reduce excessive computation complexity, we propose the complementary dynamic population size (DPS) to improve the execution speed of the classical PSO.

During the early phase, distributed positions of the particles are random and mostly are far away from the global optimum; the calculations then will make those far away particles to the region closer to the desired region. However, in this phase the algorithm process is out of order and time-consuming, and the convergence rate tends to be exponential. Nevertheless, in the latter stages, the convergence rate will be asymptotic as the algorithm approaches the exact global optimal value. To improve the execution speed, instead of the operation in the classical algorithm, a small number of particles (about 10% to 25% of the population size) are generated at the beginning, and, as the increment of loops, additional particles are created and associated with the number of iterations until the number of the particles has reached the presetting population size. Moreover, additional particles are generated and policy-based initialized, and three different methods of initialization are introduced. The first approach of initializing is to copy the best particle in POP ; the second is similar to the first one, which is to copy the best particle and update it with (1) and (2). The third way is to create one new particle randomly. When an additional particle is about to be generated, a method for initialization among the three is selected randomly. This phase of the proposed DPS-PSO is defined as Expanding Population Phase (EPP).

After EPP, the number of the particles remains, which is equal to the presetting population size. This phase of DPS-PSO is named Maintaining Population Phase (MPP).

The PSO concept prefers a larger population size to allow exploration of the widest possible region of the solution space to recognize the region containing the global optimum. Once that region is identified, the process will fine-tune the converge on the global optimum. Not as many as particles required at the beginning of the fine tuning, the search space of particles becomes much narrower in the later process. If the search space is narrow enough, the number of particles is gradually reduced by dropping lower performance particles as measured by fitness function. This phase of DPS-PSO is defined as Diminishing Population Phase (DPP). Introducing DPP into PSO process will significantly reduce the number of particles for subsequent iterations.

To conclude, components of proposed population size operator are EPP, MPP, and DPP, the invoking of which is in order, and an example of the population size operator is clearly illustrated in Figure 3.

In Figure 3, the periods between n_0 and n_1 , n_1 , and n_2 , n_2 and n_3 denote EPP, MPP, and DPP, respectively. Moreover,

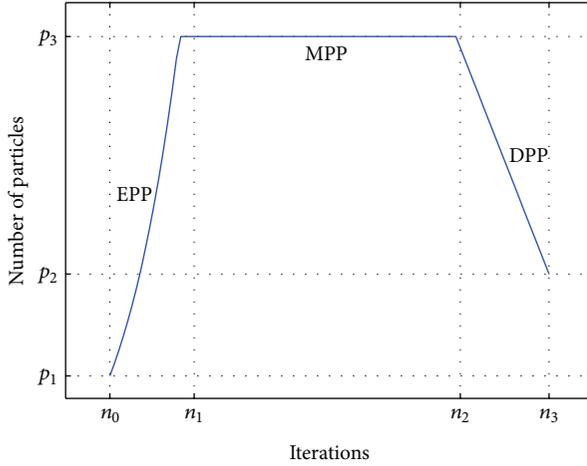


FIGURE 3: Diagram of dynamic population size operator.

p_1 , p_2 , and p_3 represent the initial, final, and presetting maximum numbers of particles, respectively.

In addition to the improvement of population size, a mutational operation is introduced to maintain the diversity of the trajectory. At the beginning of the search, the mutation operator attempts to explore all the new particles. Then, the mutation rate might be decreased with increasing iterations. Therefore, the mutation range is affected by the rate. The pseudocode of the mutation operator applied is the following.

```

% particle = particle to be mutated
% dims = number of dimensions
% currentiter = current number of iterations
% totaliter = total number of iterations
% mutrate = mutation rate
% mutrange = mutation range
function mutation_operator()
IF flip((1 - currentiter/totaliter)1/mutrate2), THEN
  whichdim = random(0, dims - 1)
  mutrange =
    (upperbound[whichdim] - lowerbound[whichdim])
    •(1 - currentiter/totaliter)1/mutrate2
  ub = particle[whichdim] + mutrange
  lb = individual[whichdim] - mutrange
  IF ub > upperbound[whichdim],
  THEN ub = upperbound[whichdim]
  IF lb < lowerbound[whichdim],
  THEN lb = lowerbound[whichdim]
  particleNEW[whichdim] = RealRandom(lb, ub)
END IF
END function

```

To verify the performance of complements, the DPS-PSO is implemented and compared with the classical PSO, EP-PSO (Expanding Population PSO), and DP-PSO (Diminishing Population PSO) [15].

Most of the algorithm parameters of the four PSO methods are configured on the same value. Additionally, the presetting population size is set to 32, which means in the classical PSO 32 particles are employed, In EP-PSO 32 will be the value of expanding result, in DP-PSO the algorithm will start with 32 particles, and in DPS-PSO the value of p_3 is 32.

To assess the performance, the four PSO approaches are applied to solve the following unconstrained maximization problem, which is introduced to assess the performance of evolutionary algorithms in [15, 16] as well:

$$\max f(x, y) = 0.5 - \frac{(\sin \sqrt{x^2 + y^2})^2 - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}. \quad (3)$$

It is worth noting that the global optimum of (3) is reached when $x = 1$ and $y = 1$.

To maintain the same experimental conditions in [15], independent optimization procedures for (3) are executed 10 times for each PSO algorithm. For each procedure, the number of iterations is set to 10000 consistently.

Table 1 shows a comparison of the execution time for the four implementations for 10 sample runs. All EP-PSO, DP-PSO, and DPS-PSO have a better execution time compared to the classical PSO. Moreover, in terms of execution time, the proposed DPS-PSO outperforms the competitions and exhibits an improvement over the classical PSO approximately 69%, EP-PSO approximately 13%, and DP-PSO approximately 10%, respectively.

Table 2 shows a convergence comparison of different implementations for the optimization problem in (3). In [15], the value of y has already been set to 1, so the search space of the experiments is unidimensional. However, the search space of the above experiments in this paper is bidimensional, which is the reason why experiments in this paper bigger numbers of iterations to converge are needed. Moreover, in terms of accuracy, the proposed DPS-PSO is better and exhibits an improvement over the classical PSO approximately 28%, EP-PSO approximately 41%, and DP-PSO approximately 11%, respectively. In terms of iterations to converge, the proposed DPS-PSO exhibits an improvement over the classical PSO approximately 68% and EP-PSO approximately 73%, respectively, which is almost similar to DP-PSO.

From the results in Tables 1 and 2, DPS-PSO reduces execution time significantly and the best results can be achieved during relatively small number of iterations.

3.3. Proposed DPS-MOPSO. In the above subsection, a modified PSO approach, known as DPS-PSO, is proposed, and the results of comparative experiments demonstrate the better performance in terms of both execution time and accuracy.

To implement green clustering, the original problem is rewritten to a multiobjective optimization (detailed in Section 4.3). MOPSO is proposed as an effective and valid

TABLE 1: Comparison of execution times.

Run	Exec. time (millisecond)				Exec. speed improvement (%)		
	PSO	EP-PSO	DP-PSO	DPS-PSO	EP-PSO	DP-PSO	DPS-PSO
1	111.87	38.21	30.20	28.94	48.31	55.48	56.60
2	104.79	34.02	36.53	29.27	55.58	53.19	60.11
3	87.23	47.70	32.02	29.11	51.08	69.06	72.39
4	91.75	35.36	34.75	29.07	62.01	62.68	68.87
5	81.00	37.23	46.74	29.14	67.94	56.20	77.93
6	81.17	37.59	40.42	29.26	67.35	63.87	77.62
7	86.82	52.91	37.95	29.37	45.32	62.56	72.45
8	98.42	48.80	52.32	29.02	44.16	40.58	64.26
9	92.67	45.21	41.36	29.25	50.77	54.93	67.99
10	86.88	35.23	35.06	29.17	65.42	65.84	72.61
Ave.	92.26	41.23	38.73	29.16	55.82	58.44	69.08

TABLE 2: Convergence comparison.

	PSO	EP-PSO	DP-PSO	DPS-PSO
Mean dist. to global optimum	0.0220	0.0268	0.0177	0.0158
Standard deviations of dist. ($\times 10^{-4}$)	1.54	1.97	1.59	1.30
Mean iterations to converge	6987	8229	1704	2230
Convergence speedup over classical PSO	—	-17.78%	75.61%	68.08%

approach for multiobjective optimization in [17] and is employed for green clustering. Moreover, MOPSO is modified with the complements of dynamic population size and mutational operators to reduce excessive repetition during the update process for each particle within each iteration and to maintain the diversity of trajectories.

The general mechanism of MOPSO is similar to PSO (already detailed in Section 3.1), but the main differences are the methods to assess particles and to determine the value of $REP[h]$ in (1). Referring to the method of assessment, the concept of Pareto Dominance is introduced in [17], and, in consideration of the value of $REP[h]$, a secondary fitness assignment based on recursively dividing up the hypercubes, where the objective values reside, is applied [18].

MOPSO is implemented as the comparison with the proposed DPS-MOPSO, and test functions and experimental parameters are set to the same values as used in [17] (the expressions of the three test functions can be found in [17] as well).

It is worth noting that the average distance to the Pareto optimal set M_1^* is defined as

$$M_1^* = \frac{1}{|Y'|} \sum_{d' \in Y'} \min \{ \|d' - \bar{d}\|^* ; \bar{d} \in \bar{Y} \}, \quad (4)$$

where $Y', \bar{Y} \subseteq Y$ are the sets of objective vectors that correspond to a set of pairwise nondominating decision vectors, $X', \bar{X} \subseteq X$, respectively, and X corresponds to the decision variables of the problem (see [19] for further details).

Table 3 shows that the proposed approach exhibits an improvement over the classical MOPSO over about 50% in terms of execution time on average, and the accuracy enhances as well. Moreover, test function 3 in [17] is much

more complex; therefore, it takes more time for each iteration within both the classical MOPSO and DPS-MOPSO.

4. Green Clustering Implementation Based on Proposed DPS-MOPSO

In the above section, DPS-MOPSO (Dynamic Population Size Multiple Objective Particle Swarm Optimization) is implemented. In this section, green clustering based on DPS-PSO (GC-DPS-MOPSO) is proposed, the framework of which is illustrated in Figure 4. The implementation of green clustering is a scheme of double-pass configuration, and the main components of core algorithm are Proc. F. and Proc. A. (short for the faster procedure and the more accurate procedure, resp.), which are concurrent and parallel. Proc. F. is introduced for the purpose of implementing green clustering with less time consuming; therefore, the branch of Proc. F. is composed of simple and efficient approaches. On the contrary, Proc. A. is applied to achieve a more accurate solution set consisting of APs; therefore, more complex methods are involved.

The precondition of the green clustering implementation which is denoted by the database of “Loc. (Location) of AP and RP (Reference Point)” in the diagram is that the complete knowledge of the physics positions and status of all APs within the WLAN are achieved. Moreover, the other parts of the framework in Figure 4 will be studied and discussed in the following subsections.

4.1. Estimation of APs' Coverage. The content in this subsection includes the parts of Simple Modeling, Complicated Modeling, Coverage Radius of AP, and Propagation Model

TABLE 3: Comparison between classical MOPSO and the proposed DPS-MOPSO.

Test function	MOPSO			DPS-MOPSO		
	1	2	3	1	2	3
Exec. time (ms)	7.6	4.6	9.8	2.9	2.1	4.5
M_1^* ($\times 10^{-4}$)	20.57	14.74	11.61	17.31	13.92	9.57
Standard deviation ($\times 10^{-4}$)	2.86	2.02	7.21	2.37	1.83	4.36

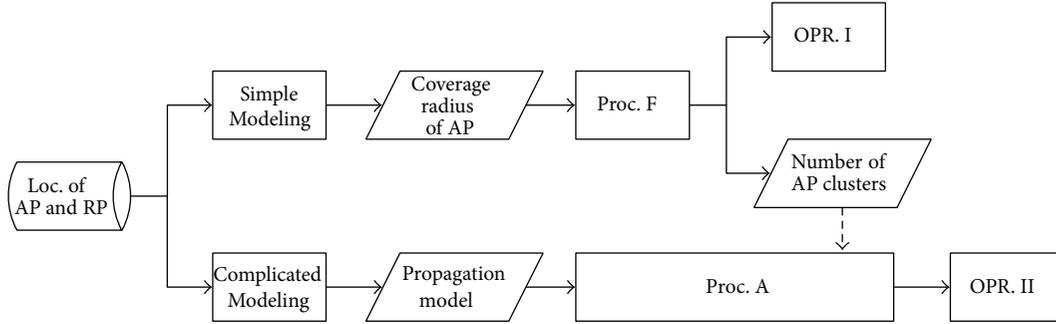


FIGURE 4: The algorithm flowchart of GC-DPS-MOPSO.

in Figure 4. The problem of green clustering is to maintain the coverage of the WLAN with less APs, and the metric of coverage is introduced in this subsection. As shown in the diagram, the core sections (Proc. F. and Proc. A.) are concurrent and parallel, and both are introduced for green clustering. Nevertheless, the designed implementation processes between Proc. F. and A. are completely different, and the major distinction is the complexity level of the algorithm within each procedure. As a consequence, the metrics of coverage for Proc. F. and A. are designed to be different.

In Proc. F., to guarantee time efficiency, a simple and less time-consuming metric of coverage is introduced, which is based on Friis free space equation. The simple metric Mtd , known as Simple Modeling, is given by the form of the maximum transmission distance

$$Mtd = 10^{((L-46)/10n)}, \quad (5)$$

where L denotes the path loss and n is the attenuation factor which differs in different environments. If the distance between a target region and an AP is greater than the value of Mtd , the region is considered as being uncovered by the current AP.

Furthermore, the metric for Proc. A. is more complex; unlike achieving a decision through a numeric comparison, an indoor propagation system based on the combinations of ray-tracing and FDTD (Finite-Different Time-Domain) techniques is introduced. In order to implement this type of estimation of APs' coverage, the region where the WLAN resides is modeled and meshed for a large amount of RPs. Additionally, TPs (Test Points) are selected among RPs to assess the coverage. To estimate the region covered by an AP, the wave propagation and penetration from the current AP are modeled within a constrained and reasonable area. If the RSS (Received Signal Strength) at a TP is greater than a presetting value which is equal to the Rx sensitivity

in accordance with some specific protocol (to be specific 802.11b is introduced here), the current TP is considered as being covered by the corresponding AP. The analogous combinations of ray-tracing and FDTD have already been proved in [20–22] and implemented in our previous work [13] as well.

4.2. The Faster Procedure. The content in this subsection includes the parts of Proc. F., OPR. I, and number of AP clusters in Figure 4. To implement green clustering in the form of Proc. F., the primal problem is adapted to a classification problem. All of the APs within the WLAN are considered as patterns to be classified and each pattern is represented by the property of physics position, and the classification is based on the known information of locations. To obtain the result fast, K -means algorithm is employed.

The analogous procedure of Proc. F. has already been studied and implemented in our previous work [13]. After applying Proc. F., a set composed of some specific APs is achieved as the center of each category and is considered as the cluster-head AP of a green cluster as well. Green clustering based on Proc. F. is implemented, and as the metric of assessing the coverage the Simple Modeling is applied as well.

Additionally, before executing K -means a target number of categories is the major prerequisite, Proc. F. starts with a bigger value (detailed in Section 5), and the number of K -means processes is diminishing. As a coarse method to implement green clustering, the results of Proc. F. (denoted by R_F) can provide Proc. A. with an upper boundary of the cluster-head APs, which is denoted by the section of number of AP clusters and the dot line downward. Besides, before applying green clustering, all the APs within the WLAN are powered on; once Proc. F. is completed, OPR. I is executed, which denote operations consisting of increasing the transmitting powers of APs in the set of R_F and then

powering off APs in R_F^C . So far, the branch of fast green clustering is implemented, and only a coarse result consisting of cluster-head APs is achieved.

Besides, it is worth noting that the proposed green clustering implementation is designed as a parallel structure for the purposes of speed and accuracy, and the faster procedure is for speed. Therefore, the above branch in Figure 4 consists of a method of Simple Modeling (expressed by (5)) and Proc. F. based on a simple algorithm (K -means). As a consequence, the results are coarse and preliminary rather than accurate or final.

4.3. The More Accurate Procedure. The content in this subsection includes the parts of Proc. A. and OPR. II in Figure 4. In Section 3, DPS-MOPSO is proposed and applied to implement the green clustering. On account of the fact that DPS-MOPSO is a method for multiobjective optimization, the primal problem of green clustering is adapted once again.

4.3.1. Problem Description. In Proc. A., a set of trial cluster-head APs is considered as a feasible solution, and the region covered by the WLAN is considered as the boundary for feasible solutions, which is the search space as well. Therefore, the solving process of green clustering can be considered as the optimization procedure to achieve the global best solution within the search space. Moreover, the objectives are prerequisite, and the problem of green clustering can be given by

$$\begin{aligned} \arg \min \quad & \left\{ \sum_{i \neq j} (S_i \cap S_j) \right\} \\ \text{s.t.} \quad & \bigcup_{i=1}^n S_i \geq S_{\text{basic}}, \end{aligned} \quad (6)$$

where S_i and S_j are the cover area of the i th AP and the j th AP, respectively, which are both in the set of a feasible solution, $i \neq j$. $\sum_{i \neq j} (S_i \cap S_j)$ denotes overlap area, and S_{basic} represents the condition of network coverage.

Each AP in the trail feasible solution is represented by the attribution of physics position expressed as Loc . Therefore, the basic problem can be adapted as a problem of multiobjective optimization expressed by

$$\begin{aligned} \text{Minimum} \quad & F = [\text{objfunc1}(Loc), \text{objfunc2}(Loc)] \\ \text{subject to} \quad & R \in \Omega, \end{aligned} \quad (7)$$

where R is a feasible solution consisting of a set of trial cluster-head Aps and $\text{objfunc1}(Loc)$ denotes the function that calculates the coverage with Loc . On account of the fact that this is a minimizing process the results $Scale$ are handled by $\text{abs}(1 - Scale)$ which means the area is not covered by R . While $\text{objfunc2}(Loc)$ represents the function that calculates the overlap area, Ω denotes the region of the WLAN, and $R \in \Omega$ means that the location of each AP in R is reasonable. It is worth noting that the assessment of coverage in Proc. A. is achieved through Complicated Modeling.

In our earlier work [13], EA is employed to solve the multiobjective optimization problem, benefiting the structure of the algorithm. EA is a stochastic search method that mimics the metaphor of natural biological evolution and models natural processes, such as the selection, recombination, mutation, migration, locality, and neighborhood. However, considering the long computing time of EA, which is not suitable for the planning function of green clustering, therefore, PSO, which is a type of EA with a simpler mechanism, is introduced to implement green clustering (detailed in Section 5.4).

4.3.2. Proc. A. Based on DPS-MOPSO. The WLAN region is meshed; RPs and TPs are generated in order. Each RP is attributed with the information of corresponding physics position to assess the coverage, and, in order to represent the results of assessment, two additional attributions are assigned for each TP, which are the properties of coverage and overlap levels. Additionally, the concept of resolution is introduced, and lower or higher resolution represents a larger or smaller number of TPs, which are selected from RPs to assess the coverage and overlap.

Proc. A. is as follows.

- (1) Initialization:
 - (a) Obtain the physical positions of all APs in the WLAN.
 - (b) Calculate and determine the original number of green clusters N and set an upper bound Nub for N .
 - (c) Set a smaller number of iterations for DPS-MOPSO expressed as $totaliter1$, a lower resolution and a smaller ratio ξ_1 which represents the uncovered area.
- (2) **WHILE** $totaliter1$ has not been reached,
 - (a) apply the DPS-MOPSO and the objective values are calculated with the low resolution and Complicated Modeling;
 - (b) increase the counter of iterations.
- (3) **END WHILE.**
- (4) Evaluate the results with ξ_1 .
 - (a) **IF** any of feasible solutions' coverage is not bigger than ξ_1 , **THEN GO TO** step 5.
 - (b) **IF** all of feasible solutions' coverage is bigger than ξ_1 , and N has not reached Nub , **THEN** $N++$ and **GO TO** Step 2.
 - (c) **IF** all of feasible solutions' coverage is bigger than ξ_1 , and N has reached Nub , **THEN** the algorithm fails. In order to avoid failure, a bigger Nub is recommended, and in the implementation of green clustering the results of Proc. F. are combined as the upper boundary.

- (5) Initialization for another DPS-MOPSO processing:
 - set a higher resolution and a bigger number of iterations to be executed for the current procedure expressed as $totaliter2$, and $totaliter2 \geq 2 \times totaliter1$ is recommended.
- (6) **WHILE** $totaliter2$ has not been reached,
 - (a) apply the DPS-MOPSO and the objective values are calculated with the high resolution;
 - (b) increase the counter of iterations.
- (7) **END WHILE**.
- (8) Set a bigger ratio ξ_2 , the value of ξ_2 is problem-specific, and $\xi_2 \leq \xi_1/5$ is recommended. With ξ_2 the feasible solutions in REP whose first objective value (calculated by $objfunc1(Loc)$) is not bigger than ξ_2 are acceptable. Once acceptable solutions are obtained, we choose the solution from the acceptable solutions which is most close to the acceptable hyperplane $ObjVal1 = \xi_2$ as the ultimate feasible solution which is denoted by R_A .

Additionally, once Proc. A. is completed, OPR. II is carried out, which denote operations of powering on APs in $R_F^C \cap R_A$ and then powering off APs in $R_F \cap R_A^C$. And after OPR. II, GC-DPS-MOPSO is implemented.

4.4. Parallel Structure. As shown in Figure 4, the proposed green clustering consists of two parallel and concurrent branches, which are based on K -means and DPS-MOPSO, respectively. Each branch has the capacity of implementing green clustering. However, the design of parallel structure is necessary rather than redundant. In Figure 4, the above branch is based on K -means, through which coarse results (more APs are needed to maintain basic network coverage) can be obtained in a relatively short time. According to the result, the central controller can power off APs. As the above branch runs, the below branch based on DPS-MOPSO is conducted as well for future power conservation. Proc. A. can achieve a more accurate result (less APs are on) at the cost of more time (than Proc. F.). In other words, to implement green clustering two branches are executed simultaneously. To be specific, the above one is to guarantee the time efficiency, and the below one is to achieve a topology with less APs for future energy efficiency.

5. Simulations

In this section, simulations are performed to verify the method of the proposed green clustering implementation GC-DPS-MOPSO. Three different experimental environments are introduced, and the algorithms in [12] and our previous work [13] are presented as comparisons.

5.1. Experiment 1. Exp. (Experiment) 1 and Exp. 2 are conducted in our previous work as well, and in consideration of comparison, the experimental conditions and environments

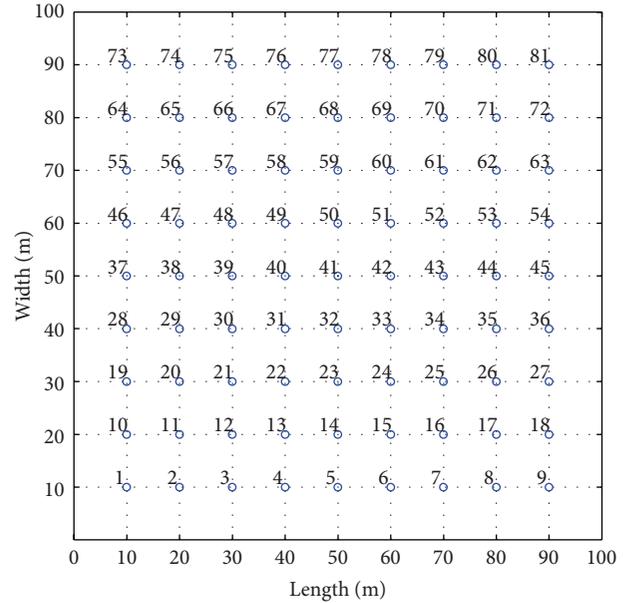


FIGURE 5: The deployment of APs in Exp. 1.

are set the same. Exp. 1 is in an ideal environment, where 81 APs are deployed uniformly in a $100\text{ m} \times 100\text{ m}$ region, as shown in Figure 5.

In GC-DPS-MOPSO, Proc. F. starts with a bigger number K of categories which can be given by empirically

$$K = \text{pow} \left[2, \frac{S_{\text{basic}}}{\pi \cdot Mtd^2} \right]. \quad (8)$$

The results of Proc. F. and A. are shown in Figure 6, which show that Proc. F. applies 9 specific APs to maintain the network coverage and 8 for Proc. A.; therefore, approximately 90% of energy is saved.

5.2. Experiment 2. Unlike Exp. 1, Exp. 2 is in a practical model. To be specific, the floor where our office resides with 27 APs is modeled to verify the proposed algorithm, as shown in Figure 7.

The results of Proc. F show that 14 APs are required to cover the interesting region, whereas 9 APs are needed to maintain the coverage according to Proc. A. (shown in Figure 8), from which we can deduce that approximately 2/3 of energy is saved under the given circumstance.

In Figure 8, a blue solid triangle denotes an AP. The coverage assessments in GC-DPS-MOPSO are based on Simple and Complicated Modeling, both of which are composed of mathematical methods for estimations; therefore, to test the performance of the proposed approach in practical model, MPs (Measuring Points) are generated in the corridors and 3 rooms with a 0.5 m sampling interval. According to the results of green clustering, at each MP we record values of RSS from all of the APs that are still on. In Figure 8 the points with colors denote MPs, the values of RSS are represented by different colors, and all selected MPs on the floor in practice are covered by the APs specified by the results of green clustering. Therefore, in Exp. 2, both results of the estimation and measurement show the validity of GC-DPS-MOPSO.

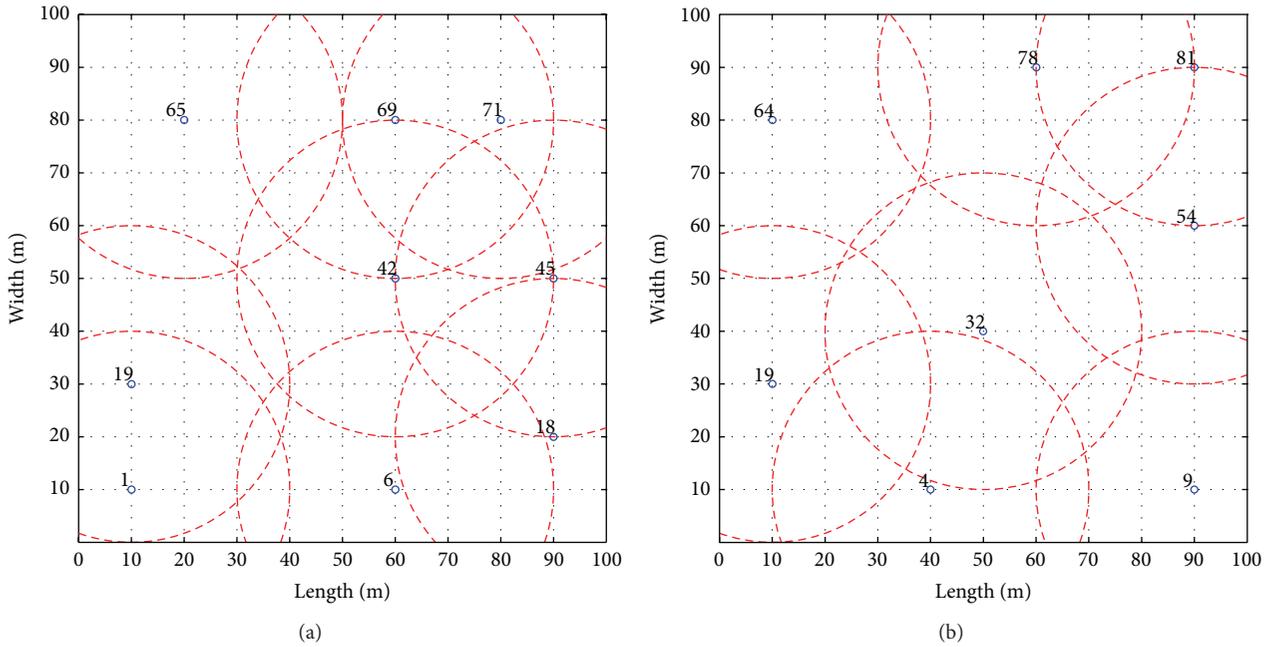


FIGURE 6: Results of Exp. 1: the former is obtained by Proc. F and the latter is achieved by Proc. A., where 9 and 8 APs are powered on, respectively.

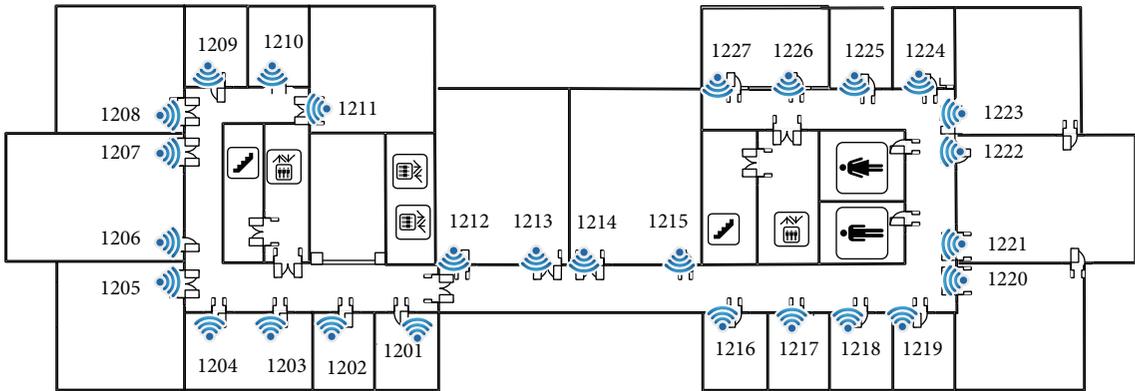


FIGURE 7: The floor layout and AP deployment in Exp. 2 where 27 APs are deployed, and the numbers denote doors.

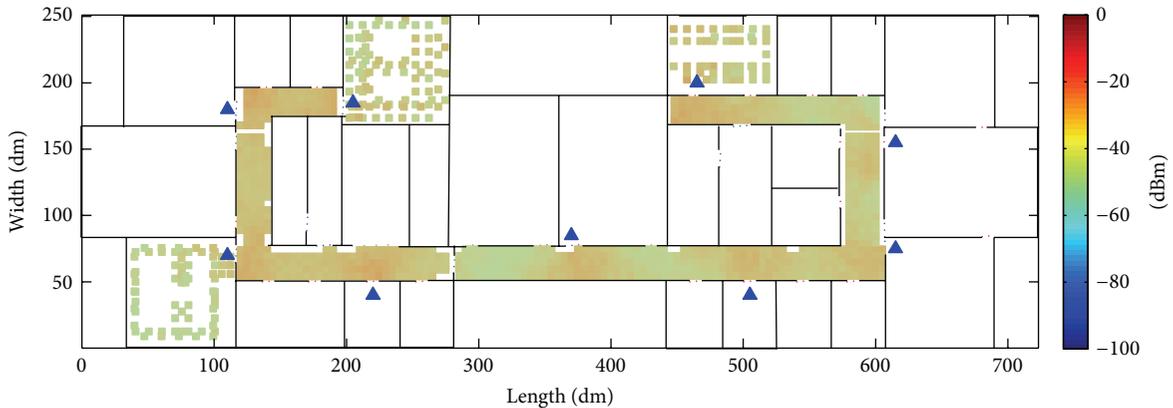


FIGURE 8: Results of Exp. 2, where 9 APs are powered on and 66.7% of energy can be saved.

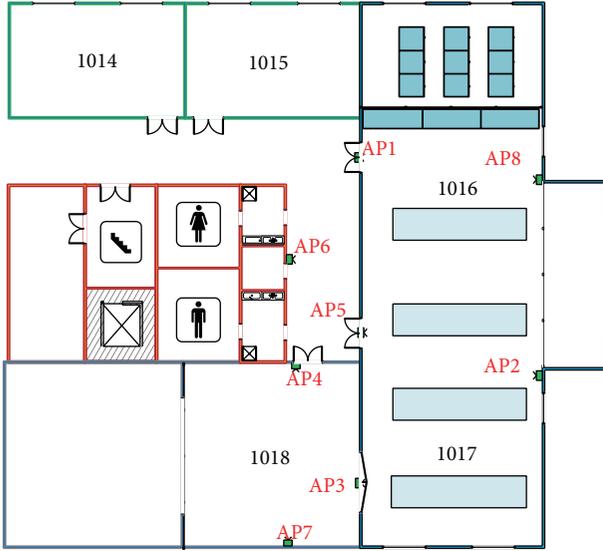


FIGURE 9: The floor layout and AP deployment in Exp. 3 where 8 APs are deployed, and the numbers denote doors.

5.3. Experiment 3. The environment in Exp. 3 is in another laboratory of ours, and the floor layout where 8 APs are deployed is modeled. Furthermore, in this experiment, a new function of GC-DPS-MOPSO is discussed and simulated. In Exp. 1 and Exp. 2, positions of APs are previously fixed, so the results of GC-DPS-MOPSO are to determine that how many and which APs are powered on; therefore, Exp. 1 and Exp. 2 prove the AP selection function. If GC-DPS-MOPSO is employed before the AP deployment, the results can be taken as reference or guidance for AP deployment and considered as the foundation of the AP deployment, which can provide the basic network coverage. Therefore, another function of GC-DPS-MOPSO is AP planning that is proved in this subsection. The floor layout and AP deployment of Exp. 3 are shown in Figure 9. Firstly, the green clustering algorithm implements the function of selection.

From Figure 9, the AP deployment on this floor is not appropriate to provide the whole experimental region with sufficient network coverage, because the APs are almost deployed in the lower right region, and the network coverage rate is only 67.04% calculated based on Complicated Model (excluding the elevator shaft filled with oblique lines in the diagram of the layout). In this case, GC-DPS-MOPSO is introduced and achieves results of AP selection, which, to be specific, are only powering on APs labeled 1, 6, and 7 is able to maintain the same network coverage as the always-on counterpart. In summary, about 62.5% of energy is saved when the traffic load is low or few end users are online.

Furthermore, for performing the AP planning function, we assume that the coverage of all areas (except for the elevator shaft) in Figure 10 is necessary. The original deployment of APs is without capacity of overall full coverage. In order to apply GC-DPS-MOPSO, the WLAN region is meshed, RPs are generated with a sampling interval of 0.05 m, and each RP represents an available location for AP deployment. To test and calculate the values of covered and overlap areas with a

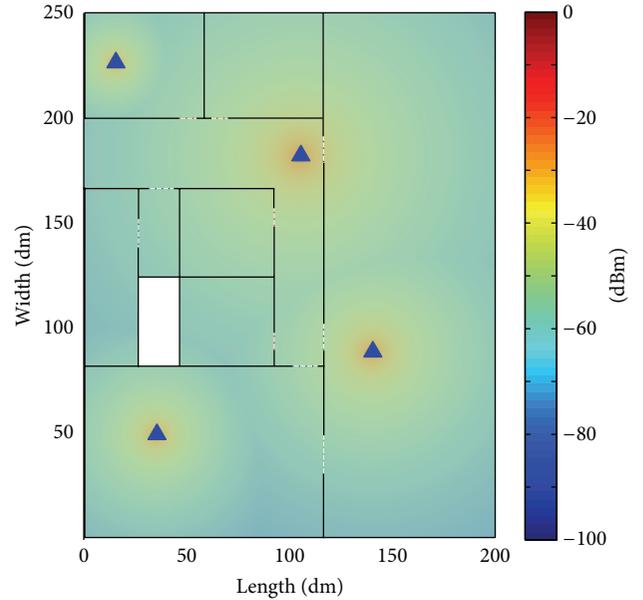


FIGURE 10: Results of the planning function in Exp. 3: 4 APs can maintain the overall network coverage, where a blue triangle denotes a location to deploy an AP, different colors represent different values of RSS, and only the maximal RSS at each TP is recorded.

lower resolution TPs are generated with a sampling interval of 0.1 m.

Referring to Proc. F., all RPs in the WLAN region are considered as patterns to be classified, and the results are employed as the upper boundary of the cluster-head AP number for Proc. A.

And, referring to Proc. A., the search space for DPS-MOPSO becomes the region composed of RPs. Therefore, within each iteration of DPS-MOPSO, the algorithm is searching for the constrained space composed of RPs, and objective values are calculated based on the locations of RPs in the feasible solutions and all TPs within a reasonable distance. To be specific, in terms of wave propagation in Exp. 3, if the distance between Rx and Tx nodes is more than 30 m, the Rx region is determined to be uncovered by the current Tx device in an indoor environment. The results of AP planning are shown in Figure 10.

In Figure 10, if the AP deployment is optimized, 4 APs at positions specified by GC-DPS-MOPSO can maintain the overall network coverage. Based on the instructional scheme, additional APs can be deployed for the purpose of meeting the peak demand, which is the designed goal of traditional WLANs; hence, it seems that the green clustering provides the foundation, and other APs enhance the WLAN with function and capacity.

5.4. Comparisons and Analysis. The proposed green clustering implementation has two functions, which are AP selection and planning. In this subsection, both functions are compared with green clustering algorithms in [12, 13]. Firstly, the AP selection function is taken for comparison.

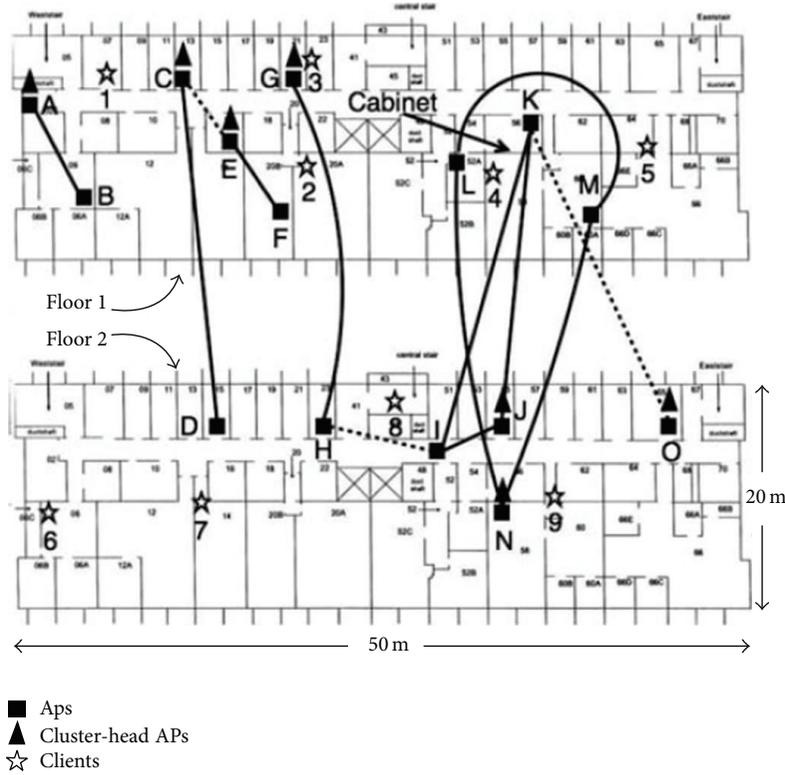


FIGURE 11: The floor layout and AP deployment of the comparison task, where, on 2 floors, each of which is 50 m × 20 m and 14 APs are deployed.

5.4.1. AP Selection. This function is implemented after AP deployment and for the purpose of choosing specific APs to maintain basic network coverage. Firstly, green clustering in [12] is introduced and the corresponding building layout and AP deployment are shown in Figure 11.

According to the environment given in Figure 11, 14 APs are deployed in the building, and 7 AP clusters are formed according to their green clustering algorithm, which means 50% of energy is saved, whereas the results of the proposed green clustering implementation are 88.89% in Exp. 1, 66.67% in Exp. 2, and 62.5% in Exp. 3. Therefore, only in terms of the saved energy ratio, our approach outperforms their algorithm.

Referring to the method in our previous work [13], in terms of the saved energy ratio, the performance is the same as that of the GC-DPS-MOPSO. The difference between the two green clustering implementations is that Proc. A. is based on EA in [13], whereas in this paper the main component of Proc. A. is DPS-MOPSO, which is more competitive in terms of less time consumption.

In consideration of fairness, a parameter which can describe the green clustering gain of algorithms in different indoor environments is introduced [13], and the parameter θ can be defined as

$$\theta = \frac{r_{\text{SAVED}}}{\mu_{\text{AP}}} \cdot n, \quad (9)$$

where r_{SAVED} denotes the ratio of saved energy, μ_{AP} is the parameter in (9) to describe the density of APs, and n

is the attenuation factor in (9) to demonstrate different environments (set to 3.5):

$$\mu_{\text{AP}} = \frac{n_{\text{AP}}}{S}, \quad (10)$$

where n_{AP} is the number of APs in the WLAN and S is the coverage of the WLAN accordingly.

Firstly, the value of θ in [12] is 125. Then, referring to the method in [13], the values of θ are 329.22 in Exp. 1 and 162.05 in Exp. 2. Finally, the values of θ are the same as those in [13] for Exp. 1 and Exp. 2 and 136.72 for Exp. 3. Therefore, in terms of either saved energy ratio or θ , the proposed method outperforms the algorithm in [12] and has the same performance as the algorithm in [13].

Proc. A. in our previous work is implemented through the modified EA. In this paper, a much more time-efficient algorithm DPS-MOPSO is employed. To prove the efficiency of our new algorithm, we test both methods in the same condition (of Exp. 2), where the population size is set to 32, and the number of total generations and iterations is set to 1000. Ten independent experiments are executed for each algorithm, and the results are shown in Table 4.

On account of the complexity of calculating coverage for Proc. A., the time consumed for each generation or iteration is high, especially when compared with the execution times in Table 3. According to Table 4, the time efficiency of the new algorithm largely improves. Although performances of saved energy ratio are the same, concerning the time efficiency (according to the concept of Pareto Dominance), the new algorithm is superior to the old one.

TABLE 4: Comparison of execution time.

Run	Mean execution time (s)	
	The old	The new
1	2.3273	0.3650
2	2.5155	0.8035
3	2.2572	0.5725
4	2.2773	0.4937
5	2.4986	0.5715
6	2.8065	0.4795
7	2.3461	0.4876
8	2.5743	0.6490
9	2.4549	0.6409
10	2.7235	0.6417
Ave.	2.4781	0.5705

5.4.2. AP Planning. In terms of guidance for AP deployment, Liao et al. propose an optimal set under the diamond pattern to achieve optimal coverage [23]. In [24], a scheme named concentric-hexagon-oriented multichannel assignment is proposed to arrange APs as concentric-hexagon for solution of the interference problem. Reference [25] concentrates on AP placement problem in green wireless cooperative communication networks. In [26], a coarse deployment can be achieved based on Euclidean distance. Additionally, genetic algorithm is employed for AP deployment in [27]. However, all of the works simply implement the AP deployment based on a fixed value of distance (similar to Simple Modeling of GC-DPS-MOPSO), and none involves wave propagation and penetration in indoor environments.

In [28, 29], schemes of AP deployment are studied and discussed under more complex conditions, but the focus between their work and ours is different. To be specific, the research in [28] focuses on avionics equipment. Reference [29] studies mobility model of end users, and their AP deployment algorithm is based on the topology of mobility graphs.

The method in neither [12] nor [13] involves the function of planning for the AP deployment; the method in [12] is functional based on the existing APs, but it is incapable of AP planning, whereas the scheme of the method in our previous work is similar to GC-DPS-MOPSO. Therefore, theoretically the method in [13] can be used for AP planning as well. In order to implement AP planning, as operations in Exp. 3, the WLAN region is needed to be meshed firstly, and then RPs and TPs are generated with different sampling interval. However, in practice, we find that when implementing the AP planning function, the execution time of our modified EA seems to be unacceptable. Therefore, in terms of planning the AP deployment, the previous algorithm is not suitable. To conclude, among all the given methods, GC-DPS-MOPSO outperforms others in terms of the AP planning ability.

6. Conclusion

High-density WLANs are deployed in increasing numbers of enterprise offices and college campuses at present. To ensure

the network coverage, all of the APs in the WLANs have to be powered on, which causes severe energy wastage. In this paper, the energy-efficient strategy is discussed and a new green clustering implementation, which is an innovative algorithm, is proposed.

The proposed green clustering is composed of two parallel and concurrent procedures, which are known as Proc. F. and A. In terms of the different procedures, the problem descriptions of green clustering are based on different models. Referring to Proc. F., the green clustering is rewritten to a classification problem, and K -means algorithm is employed to implement the green clustering as a fast processing. With regard to Proc. A., the original problem is described as a multiobjective optimization problem. To achieve more accurate results, we modified PSO algorithm and proposed an approach, known as DPS-MOPSO, to form green clusters. Based on the framework of the classical MOPSO, a dynamic population size and mutational operators are proposed and applied as complements, and the new DPS-MOPSO outperforms the classical method in terms of both execution times and accuracy. Moreover, on account of the fact that DPS-MOPSO is a more time-efficient method, especially when compared with EA, a new function which is AP planning can be implemented. The traditional green clustering algorithms are usually based on the existing APs to form green clusters, which can be considered as AP selection. The new function works as a guidance or reference for AP deployment.

Finally, in Section 6, simulations of the proposed green clustering implementation GC-DPS-MOPSO are clearly presented under three different experimental conditions, where the AP selection and planning functions are both proved. Moreover, the comparisons with two existing green clustering algorithms are performed as well. The results show that GC-DPS-MOPSO can save approximately 60% up to 90% energy; meanwhile the coverage is maintained during periods when few users are online or when the traffic load is low. The proposed GC-DPS-MOPSO outperforms the method in [12] in terms of energy-saving ratio and is superior to the approach in our previous work [13] in terms of time efficiency.

The green clustering algorithm is proposed as the first method for energy-efficient strategy in high-density WLANs in this paper. The algorithm will be foreshadowing for future study energy-efficient strategy, in which the optimization for transmitting power of cluster-head APs will be concerned. Additionally, GC-DPS-MOPSO will be supplemented comprehensively further. After green clustering, the mechanism of estimating user demands based on Markov Chain Model and queuing theory will also be studied, and access selection scheme based on IEEE 802.11 PCF (Point Coordination Function) for energy efficiency in high-density WLANs will be proposed in the near future.

Conflict of Interests

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

Acknowledgments

This research is supported by the National Science Foundation (Grant no. 61101122), National Science and Technology Major Project (Grant no. 2012ZX03004-003), and National High Technology Research and Development Program of China (863 Program) (Grant no. 2012AA120802).

References

- [1] "Aruba selected by Microsoft for next generation wireless LAN," 2005.
- [2] <http://muniwifi.org/>.
- [3] A. P. Jardosh, G. Iannaccone, K. Papagiannaki, and B. Vinnakota, "Towards an energy-star WLAN infrastructure," in *Proceedings of the 8th IEEE Workshop on Mobile Computing Systems and Applications (HOTMOBILE '07)*, vol. 15, pp. 85–90, February 2007.
- [4] S. D'Alessandro, N. Moret, and A. M. Tonello, "Green hybrid FMT for WLAN applications," in *Proceedings of the Wireless Days (WD '10)*, pp. 1–5, October 2010.
- [5] V. Miliotis, A. Apostolaras, T. Korakis, Z. Tao, and L. Tassiulas, "New channel allocation techniques for power efficient WiFi networks," in *Proceedings of the IEEE 21st International Symposium on Personal, Indoor and Mobile Radio Communications Workshops (PIMRC '10)*, pp. 347–351, Istanbul, Turkey, September 2010.
- [6] J. Rabaey, "The standby power challenge: wake-up receivers to the rescue," in *Proceedings of the International Symposium on VLSI Technology, Systems, and Applications (VLSI-TSA '09)*, p. 42, Hsinchu, Taiwan, April 2009.
- [7] M. A. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "A simple analytical model for the energy-efficient activation of access points in dense WLANs," in *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (ICEEC-NE '10)*, pp. 159–168, April 2010.
- [8] Y. Chen, Q. Yan, J. Yin, and X. Chai, "Power-efficient access-point selection for indoor location estimation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 7, pp. 877–888, 2006.
- [9] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Kernel-based positioning in wireless local area networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 689–705, 2007.
- [10] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Intelligent dynamic radio tracking in indoor wireless local area networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 3, pp. 405–419, 2010.
- [11] S.-H. Fang, T.-N. Lin, and P. Lin, "Location fingerprinting in a decorrelated space," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 685–691, 2008.
- [12] A. P. Jardosh, K. Papagiannaki, E. M. Belding, K. C. Almeroth, G. Iannaccone, and B. Vinnakota, "Green WLANs: on-demand WLAN infrastructures," *Mobile Networks and Applications*, vol. 14, no. 6, pp. 798–814, 2009.
- [13] Y. Lu, X. Tan, Y. Mo, and L. Ma, "A new green clustering algorithm for energy efficiency in high-density WLANs," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 2, pp. 326–354, 2014.
- [14] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, December 1995.
- [15] B. Soudan and M. Saad, "An evolutionary dynamic population size PSO implementation," in *Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA '08)*, pp. 1–5, April 2008.
- [16] L. Davis, *Handbook of Genetic Algorithm*, Van Nostrand Reinhold, 1991.
- [17] C. A. Coello Coello and M. S. Lechuga, "MOPSO: a proposal for multiple objective particle swarm optimization," in *Proceedings of the Congress on Evolutionary Computation (CEC '02)*, pp. 1051–1056, May 2002.
- [18] J. D. Knowles and D. W. Corne, "Approximating the non-dominated front using the Pareto archived evolution strategy," *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.
- [19] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.
- [20] M. Thiel and K. Sarabandi, "A hybrid method for indoor wave propagation modeling," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 8, pp. 2703–2709, 2008.
- [21] Y. Wang, S. Safavi-Naeini, and S. K. Chaudhuri, "A hybrid technique based on combining ray tracing and FDTD methods for site-specific modeling of indoor radio wave propagation," *IEEE Transactions on Antennas and Propagation*, vol. 48, no. 5, pp. 743–754, 2000.
- [22] Y. Wang, S. K. Chaudhuri, and S. Safavi-Naeini, "An FDTD/Ray-tracing analysis method for wave penetration through inhomogeneous walls," *IEEE Transactions on Antennas and Propagation*, vol. 50, no. 11, pp. 1598–1604, 2002.
- [23] L. Liao, W. Chen, C. Zhang, L. Zhang, and W. Jia, "Wireless access point deployment for both coverage and localization," in *Proceedings of the 5th IEEE International Conference on Networking, Architecture and Storage (NAS '10)*, pp. 348–356, July 2010.
- [24] F.-Y. Leu and P.-S. Liu, "A channel assignment and AP deployment scheme for concentric-hexagon based multi-channel wireless networks," in *Proceedings of the 13th International Conference on Network-Based Information Systems (NBIS '10)*, pp. 504–509, September 2010.
- [25] X. Zhang, Z. Zheng, J. Liu, X. Shen, and L.-L. Xie, "Optimal power allocation and AP deployment in green wireless cooperative communications," in *Proceedings of the IEEE Global Communications Conference (GLOBECOM '12)*, pp. 4000–4005, December 2012.
- [26] Y. He, W. Meng, L. Ma, and Z. Deng, "Rapid deployment of APs in WLAN indoor positioning system," in *Proceedings of the 6th International ICST Conference on Communications and Networking in China (CHINACOM '11)*, pp. 268–273, August 2011.
- [27] C.-S. Wang and L.-F. Kao, "The optimal deployment of Wi-Fi wireless access points using the genetic algorithm," in *Proceedings of the 6th International Conference on Genetic and Evolutionary Computing (ICGEC '12)*, pp. 542–545, August 2012.
- [28] C. Zhang, J. Yu, and K. Pang, "Multiple access points deployment optimization in cabin wireless communications," *IEEE Antennas and Wireless Propagation Letters*, vol. 12, pp. 1220–1223, 2013.
- [29] T. Wang, W. Jia, G. Xing, and M. Li, "Exploiting statistical mobility models for efficient Wi-Fi deployment," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 1, pp. 360–373, 2013.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

