Minimum-cost Deployment of Adjustable Readers to Provide Complete Coverage of Tags in RFID Systems

You-Chiun Wang and Shu-Ju Liu

Abstract—In Internet of things (IoT), radio frequency identification (RFID) plays an important role to help people rapidly obtain information of objects associated with tags. Passive tags are cheap and require no batteries to operate, so they are widely used in RFID applications. Readers, on the other hand, have to provide power to activate passive tags to get their data. However, collision occurs when two readers send signals to a tag at the same time. Therefore, it is critical to decide the locations of readers, namely reader deployment, to avoid collision. This paper considers adjustable readers, whose transmitted power is configurable to provide different communication range, and proposes a *minimum-cost RFID reader deployment (MR2D) problem*. Given the positions of tags, it determines how to deploy readers and adjust their transmitted power to cover all tags, such that we can use the minimum number of readers and save their energy. To facilitate data transmission and reduce hardware cost, we restrict the number of tags that each reader can cover and allow readers to have few overlapped tags in their communication range. Then, we develop an efficient solution to the MR2D problem by clustering tags into groups and placing a reader to cover each group to meet the above conditions. Simulation results show that our proposed solution not only saves the number of RFID readers but also reduces their energy consumption, as compared with existing methods.

Index Terms—collision, coverage, passive tag, reader deployment, RFID.

1 Introduction

RADIO frequency identification (RFID) is a non-line-of-sight communication technology to replace bar and quick response (QR) codes, which consists of tags and readers. Tags are associated with objects to store their information, while readers query the tags in their communication range to get these data. Generally speaking, tags are classified into two categories: *active* and *passive*. Active tags have embedded batteries, so they can transmit data to the readers on their own. Passive tags require a reader to provide electromagnetic signals as a power source to trigger them. Thus, they do not need extra batteries. Passive tags are more popular than active ones due to their low cost and infinite lifetime [1]. RFID has many applications in industry and commerce, and it is also a key technology in Internet of things (IoT) [2].

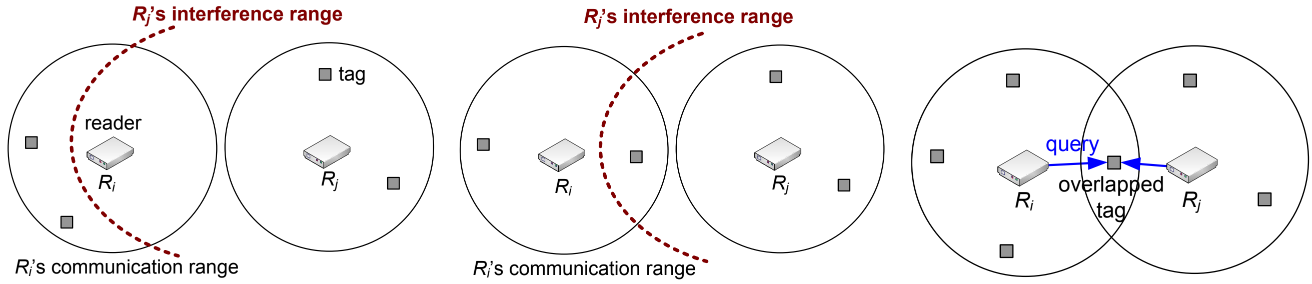
In RFID systems, there are two common scenarios of collision, namely *tag collision* and *reader collision*. Tag collision occurs when a reader sends a query to the tags in its communication range, and some of these tags respond to the reader at the same time. It will inevitably cause data collision at the reader. There are two categories of methods developed to deal with the tag collision problem [3]. *Aloha-based methods* allow each tag to randomly pick one time slot to reply its data. *Tree-based methods* build a binary tree based on tags' IDs, and the reader traverses the tree to determine which tag(s) can send replies in different time slots. Apparently, when a reader possesses many tags in its communication range, these methods have to select more collision-free time slots for data transmission.

There are two types of reader collision indicated in [4], as Fig. 1 illustrates. The *reader frequency interference (RFI) problem* assumes that each reader has interference range larger than its communication range. The RFI problem occurs when one reader R_i or its tags locate inside the interference range but outside the communication range of another reader R_j. In this case, R_j's signal could interfere with R_i or its tags. However, data collision may not necessary take place, because R_i's signal is relatively larger than R_j's signal within R_i's communication range. In addition, the RFI problem can be efficiently solved by reducing the transmitted power of a reader, as the effect of interference range significantly decreases accordingly [5]. On the other hand, the *multiple readers to tag interference (MRTI) problem* occurs when a tag locates in the communication range of two readers, and both readers simultaneously send their queries to the tag. The MRTI problem is more serious than the RFI problem, because both readers generate signals with similar strength and thereby cause data collision at the tag. Therefore, some methods [6], [7] adopt the concept of time-division multiple access (TDMA) to solve the MRTI problem, which asks these readers to select different time slots for sending queries to their overlapped tags.

The issue of reader deployment not only decides system cost but also affects performance of the above Aloha-based, tree-based, and TDMA methods. In particular, readers are much more complicated and expensive than tags [8], so they play a decisive role in the cost to construct an RFID system. Thus, it is uneconomical to place a lot of readers where most tags can be actually covered by a small subset of these readers. On the contrary, if we place too few readers such that every reader has to cover numerous tags or they have many overlapped tags, the response time of the RFID system will substantially increase, because existing methods have to find out more time slots for data transmission.

Based on the above motivation, we propose a *minimum-cost RFID reader deployment (MR2D) problem* that considers

The authors are with the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, 80424, Taiwan. E-mail: ycwang@cse.nsysu.edu.tw; m033040036@student.nsysu.edu.tw



(a) the RFI problem (case 1) (b) the RFI problem (case 2) (c) the MRTI problem

Fig. 1: Two types of reader collision.

adjustable readers whose transmitted power is configurable to provide different levels of communication range. Given the positions of tags, the MR2D problem decides the location and communication range of each reader such that the overall system cost is minimized. In particular, the cost includes not only the number of readers deployed but also the amount of their energy consumption. To guarantee the short response time of the RFID system, MR2D allows each reader to cover at most β tags and the number of overlapped tags owned by two readers is below a threshold γ, where both β and γ are configurable and depend on the application requirement.

In this paper, we show that the MR2D problem is NP-complete, and develop an efficient four-stage algorithm. In the beginning, we divide tags into groups according to the maximum communication range of a reader. Then, we adaptively adjust these groups and determine the location of a reader to be placed to cover the tags in each group, under the restriction of β. Finally, we reorganize those tags covered by multiple readers to meet the γ constraint. Experimental results by simulations demonstrate that our proposed MR2D algorithm not only significantly saves the number of readers but also reduces the overall energy consumption as comparing with other reader deployment methods, under different distributions of tags.

We give the outline of this paper as follows: Section 2 presents related work. Section 3 defines the MR2D problem while Section 4 proposes our algorithm. In Section 5, we discuss simulation results. Finally, Section 6 concludes this paper and gives future work.

2 Related Work

In the literature, the common objective of RFID reader deployment is to provide complete coverage of tags [9]. To do so, several studies use grid partition to determine the locations of readers. For example, Oztekin et al. [10] divide the sensing field into square grids and search for the grids that contain tags. They then iteratively place a reader to cover the maximum number of grids with tags. The work of [11] proposes a honey-grid deployment strategy, where the grid length is equal to the communication range of a reader and readers are deployed on the vertices of grids. It then turns off those readers whose communication range contains no tags. The study of [12] first deploys readers in a hexagon-like fashion [13], and then adopts a genetic algorithm to select a subset of readers to cover each tag. However, since the locations of readers are limited by the grid structure (e.g., grid vertices), the above studies may not use the minimum number of readers to cover all tags.

Numerous research efforts assume that the sensing field is fully covered by RFID readers and seek to eliminate (e.g., turn off) the redundant ones. In particular, when all tags in the communication range of a reader can be covered by other readers, this reader is viewed as redundant. The work of [14] iteratively picks a reader whose communication range contains the most number of tags, until all tags are covered by the selected readers. In [15], each reader broadcasts a message to all tags inside its communication range. Once a tag receives the message and has not associated with any reader yet, the tag will associate with that reader. In this way, a reader without associated tags will be redundant. Ali et al. [16] propose a *neighbor and tag elimination (NTE)* method by greedily selecting the reader with the maximum weight, where the weight of a reader depends on the number of tags that it covers and the number of its neighboring readers. In [17], readers compete to activate tags based on a greedy rule in a distributed manner. After the competition, a reader will deactivate itself if it does not have any active tags. Rashid et al. [18] partition the sensing field into grids and adopt a cellular automaton [19] to find out redundant readers, where the cellular automaton consists of a collection of cells on a grid whose states will evolve according to the states of its neighboring cells. Nevertheless, these research efforts do not consider data collision caused by readers.

Tang et al. [20] develop a schedule to activate different readers in each time slot, so as to increase the number of covered tags while avoiding potential data collision. However, they assume that all readers have the same communication range. In [21], Liu et al. propose a *reader-coverage collision avoidance arrangement (RCCAA)* problem by considering adjustable readers with multiple communication range. RCCAA gives an upper bound on the number of tags that each reader can cover, which is NP-complete. To solve the RCCAA problem, Liu et al. develop a heuristic whose idea is based on the *maximum-weight independent set* [22] to find out redundant readers and determine their communication range. Comparing with RCCAA, our MR2D algorithm has two major advantages. First, RCCAA is based on the assumption that the sensing field has been fully covered by arbitrarily deployed readers, while MR2D calculates the locations in the sensing field to deploy readers. Thus, users can precisely determine how many readers are required to construct the RFID system by MR2D. Second, RCCAA prevents two readers from covering the same tag to avoid reader collision. On the contrary, MR2D allows readers to have a limited number γ of overlapped tags, because reader collision can be easily solved by existing protocols [6], [7]. The parameter γ is configurable, so MR2D is more flexible than RCCAA. Experimental results in Section 5 will also show that our MR2D algorithm not only uses fewer readers but also saves more energy than the RCCAA method.

3 Problem Formulation

We are given a set $\mathcal{T}$ of passive tags with known positions in the sensing field[1], where each tag $t_i \in \mathcal{T}$ is modeled by one single point on the two-dimensional plane. On the other hand, we employ the binary communication model, where the communication range of readers is treated as unit disks [20], [21], [24]. Through emitting different levels of transmitted power, each reader possesses different communication radiuses of $\{r_1, r_2, \cdots, r_\alpha\}$, where $r_1 < r_2 < \cdots < r_\alpha$, and the reader can choose only one radius for communication[2]. A tag is said to be *covered* by a reader if it locates inside the communication range of that reader. Moreover, a tag is called *overlapped* if it is covered by more than one reader. In this case, two readers are *conflicting* if they have the same overlapped tags.

Then, the MR2D problem asks how to find the locations of readers in the sensing field and determine their communication radiuses to cover each tag in $\mathcal{T}$, such that the number of deployed readers is minimized and the aggregate amount of energy consumption by readers is minimized, under the following three constraints:

- β **constraint:** Each reader is able to cover at most β tags.
- γ **constraint:** Any two readers have no more than γ overlapped tags.
- ε **constraint:** A reader has at most ε conflicting neighbors.

As mentioned earlier in Section 1, the solutions to the tag collision problem have to find more collision-free time slots when a reader covers more tags. Thus, we use the β constraint to help restrict the number of time slots required by a reader to communicate with its tags. On the other hand, with both the γ and ε constraints, we can keep a short response time for the RFID system in the MRTI case of reader collision.

Let $\hat{\mathcal{R}}$ be the set of readers found in the MR2D problem. Suppose that each reader $R_i \in \hat{\mathcal{R}}$ selects a radius $r_{i,j}$ for communication. We adopt the function $E(r_{i,j})$ to represent the amount of transmitted energy spent by R_i with the communication radius $r_{i,j}$. Also, let $\Phi(R_i)$ denote the set of tags covered by R_i (with radius of $r_{i,j}$). Besides, we use the notation $|\cdot|$ to denote the number of elements in a set. Then, the MR2D problem can be formulated as an integer linear programming problem:

$$\text{minimize} \quad |\hat{\mathcal{R}}|, \tag{1}$$

$$\text{minimize} \quad \sum\nolimits_{R_i \in \hat{\mathcal{R}}} E(r_{i,j}), \tag{2}$$

$$\text{subject to} \quad \bigcup\nolimits_{R_i \in \hat{\mathcal{R}}} \Phi(R_i) = \mathcal{T}, \tag{3}$$

$$1 \le |\Phi(R_i)| \le \beta, \qquad \forall R_i \in \hat{\mathcal{R}}, \tag{4}$$

$$0 \le |\Phi(R_i) \cap \Phi(R_j)| \le \gamma, \quad \forall R_i, R_j \in \hat{\mathcal{R}}, R_i \ne R_j, \tag{5}$$

$$0 \le \sum\nolimits_{R_j \in \hat{\mathcal{R}} - \{R_i\}} I(R_i, R_j) \le \varepsilon, \forall R_i \in \hat{\mathcal{R}}, \tag{6}$$

$$\beta \in \mathbb{N}, \gamma \in \mathbb{Z}_0^+, \varepsilon \in \mathbb{Z}_0^+, \tag{7}$$

$$\varepsilon\gamma \le \beta. \tag{8}$$

Eqs. (1) and (2) are the objectives, which seek to find a minimum set of $\hat{\mathcal{R}}$ and allow readers to consume the least amount of energy, respectively. Eqs. (3)–(8) are constraints. In particular, Eq. (3) indicates that all tags in $\mathcal{T}$ have to be covered by readers. Eq. (4) implies that every reader must cover at least one tag and the maximum number of tags that it can cover is restricted by the β constraint. Eq. (5) points out that any two readers may have zero to γ overlapped tags. In Eq. (6), we adopt an indicator $I(R_i, R_j)$ to check whether two readers R_i and R_j have overlapped tags. If so, $I(R_i, R_j) = 1$; otherwise, $I(R_i, R_j) = 0$. Thus, Eq. (6) gives the ε constraint. Then, Eq. (7) limits the values of β, γ, and ε to positive integers, non-negative integers, and non-negative integers, respectively. Finally, any two readers can have at most γ overlapped tags and each reader has no more than ε conflicting neighbors, so the maximum number of overlapped tags owned by a reader is $\varepsilon\gamma$. Undoubtedly, each overlapped tag of a reader must be also a tag covered by that reader. Due to the β constraint, these $\varepsilon\gamma$ overlapped tags should not exceed the maximum number of tags (i.e., β) that the reader is allowed to cover. Therefore, $\varepsilon\gamma \le \beta$ holds in Eq. (8). In Theorem 1, we show that the MR2D problem is NP-complete.

Theorem 1. *The MR2D problem is NP-complete.*

Proof. We first show that the MR2D problem belongs to the NP class. Given an MR2D problem instance and a reader deployment result, we can verify whether the deployment result satisfies the β, γ, and ε constraints in polynomial time. Thus, the argument is proved.

Then, we show that the MR2D problem is NP-hard. As discussed earlier in Section 2, the RCCAA problem asks how to use the minimum number of adjustable readers to cover tags subject to the condition that each reader can cover at most β tags and there is no reader collision. The RCCAA problem is shown to be NP-complete in [21] and it is obvious that by setting $\varepsilon = 0$ (in this case, γ must be also zero), the MR2D problem will degenerate into the RCCAA problem. Consequently, the RCCAA problem is in fact an instance of the MR2D problem, which verifies our argument.

With the above two arguments, we prove that the MR2D problem is NP-complete. □

4 The Proposed MR2D Algorithm

Fig. 2 shows the flow chart of our MR2D algorithm, which consists of four stages to find a set of readers and adjust their communication range, such that each tag can be covered by reader(s) under the β, γ, and ε constraints.

- **Stage 1–tag classification:** In the beginning, we classify all tags in $\mathcal{T}$ into multiple subsets according to their distribution in the sensing field, where any two subsets have no overlapped tags.
- **Stage 2–tag clustering:** For each subset of tags, we further cluster them into groups such that every group contains at most β tags and these tags locate within the communication range of one reader.

1. This assumption is essential to most reader deployment methods [10]–[12], [14], [16], [18], [20], [21] discussed in Section 2. In fact, some applications adopt a 'static' tag placement strategy, for example, using tags to define the boundaries of buildings or placing tags on roads to assist the disabled [9]. In these applications, the positions of tags can be obtained a priori. Moreover, reference [23] also surveys some localization techniques to find the positions of tags.

2. Some products of RFID readers [25] support this communication property.

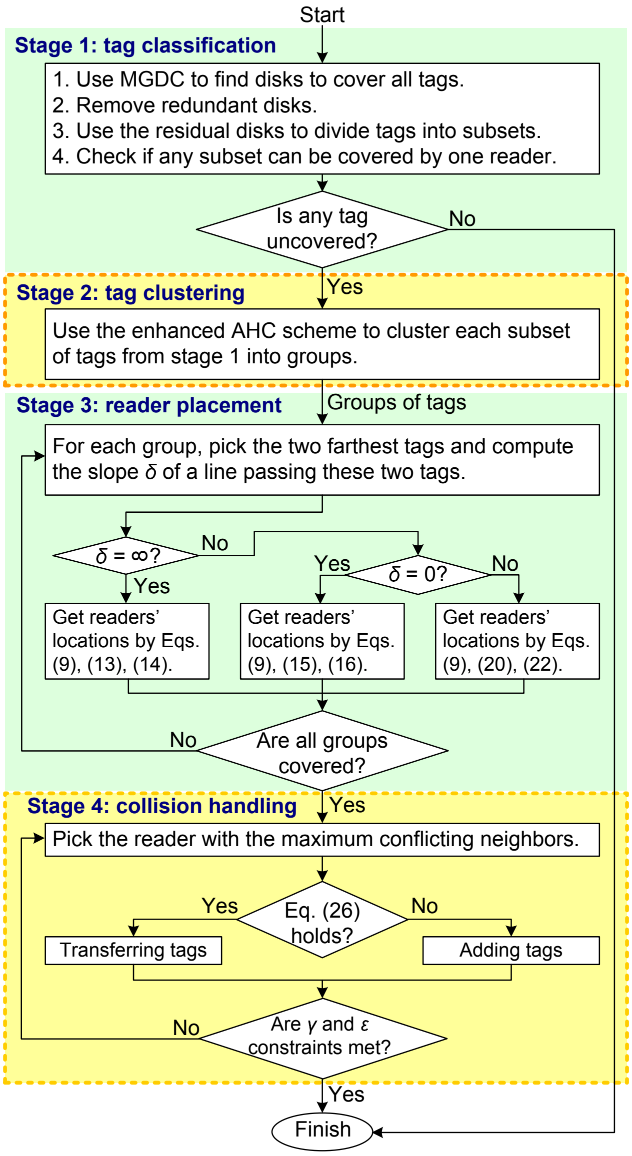


Fig. 2: The flow chart of our MR2D algorithm.

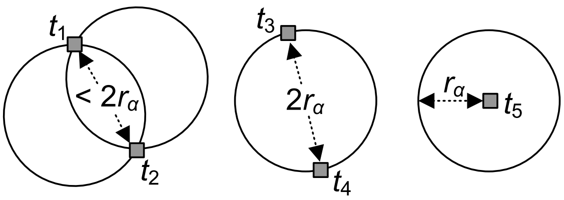


Fig. 3: Three MGDC examples.

- **Stage 3–reader placement:** Afterwards, for each group of tags, we determine the location to place a reader such that the reader can cover all tags in the group.
- **Stage 4–collision handling:** Finally, we reorganize those groups of tags whose readers are conflicting with each other, so as to satisfy the constraints of γ and ε.

Below, we present the detailed design in each stage, followed by the discussion of our MR2D algorithm.

4.1 Stage 1–Tag Classification

This stage gives a preliminary classification of tags in $\mathcal{T}$ by partitioning them into *independent* subsets, where we can

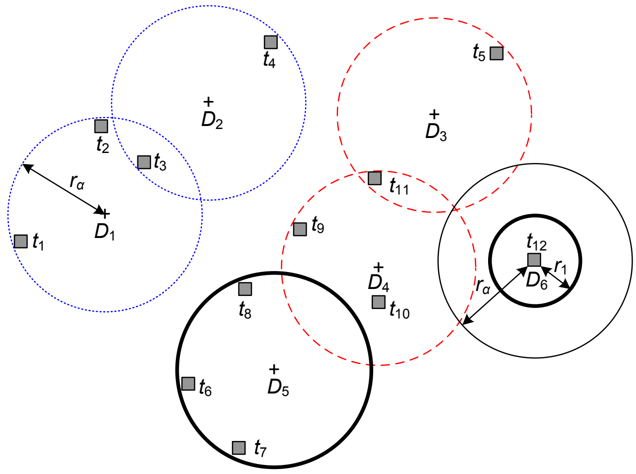


Fig. 4: An example of tag classification.

place reader(s) to cover each subset without affecting others. According to the flow chart in Fig. 2, stage 1 includes four major steps. In the first step, our idea is to use a disk with radius r_α to model the maximum communication range of one reader, and adopt such disks to cover all tags. In particular, we employ the *modified geometric disk covering (MGDC)* scheme [26] to find a set of disks to cover each tag, which possesses three rules below:

- **Rule 1:** Suppose that the distance between two tags is smaller than $2r_\alpha$. Then, we place two disks whose circumferences can intersect at the two tags. Tags t_1 and t_2 in Fig. 3 together give an example. How to calculate the coordinates of the centers of these two disks can refer to the discussion of Fig. 6(b) in Section 4.3.
- **Rule 2:** When two tags have a distance equal to $2r_\alpha$, we place one disk such that these two tags are located on its circumference. Tags t_3 and t_4 in Fig. 3 illustrate an example. The coordinates of the disk's center will be $((x_a + x_b)/2, (y_a + y_b)/2)$, where (x_a, y_a) and (x_b, y_b) are the coordinates of these two tags.
- **Rule 3:** If every neighbor of a tag has a distance larger than $2r_\alpha$, we place a disk whose center has the same coordinates with that tag. Tag t_5 in Fig. 3 shows an example.

The MGDC scheme will find out at most $O(n^2)$ disks, where n is the number of tags in $\mathcal{T}$. However, some disks may be redundant, so the second step seeks to eliminate such disks from the MGDC result. To do so, we iteratively select the disk which contains the maximum number of unmarked tags, and mark these tags, until every tag becomes marked. The remaining disks will be redundant and we can remove them accordingly.

In the third step, we divide $\mathcal{T}$ into independent subsets by the following method:

- Pick one disk, say, D_i. If D_i has some tags also located in a subset S_j, then we add all tags in D_i to S_j.
- Otherwise, we create a new subset S_i that contains all tags in D_i.
- The above operation is repeated until all disks have been checked.

Fig. 4 presents an example, where four subsets are found: $S_1 = \{t_1, t_2, t_3, t_4\}$ (i.e., D_1 and D_2), $S_2 = \{t_5, t_9, t_{10}, t_{11}\}$ (i.e.,

D_3 and D_4), $S_3 = \{t_6, t_7, t_8\}$ (i.e., D_5), and $S_4 = \{t_{12}\}$ (i.e., D_6). Apparently, because each disk represents the maximum communication range of a reader and any two subsets of tags are disjoint, we are able to 'independently' place readers for each subset without interfering with the tags in other subsets.

Then, the last step determines whether it is possible to directly place one reader to cover all tags in a subset by checking two conditions: 1) the subset contains only one disk and 2) there are no more than β tags in the subset. In Fig. 4, subsets S_3 and S_4 satisfy both conditions (where $\beta = 4$ in the example), so we can individually place one reader at the center of disks D_5 and D_6 to completely cover their tags accordingly. In addition, for each of such disks, we use as small communication range as possible for the reader to save energy (and also reduce the effect of its interference range), under the premise that all tags in the subset can be still covered by the reader. Disk D_6 in Fig. 4 gives an example, where the bold line indicates the new communication range of the reader (with radius of r_1). Afterwards, we remove those subsets of tags already placed with readers (i.e., S_3 and S_4) from $\mathcal{T}$. In case that $\mathcal{T}$ becomes empty, it means that we have found a set of readers that cover all tags while satisfying the β, γ, and ε constraints (since there are no conflicting readers), so the MR2D algorithm can finish, as shown in Fig. 2.

Remark 1. *The objective of stage 1 is to find disjoint groups of disks to cover all tags, where each group of disks indicate a subset of tags that can be placed with readers independently (i.e., without causing collision to the tags in other subsets). We can provide a 'rough' classification of tags in this stage, because each subset of tags will be further clustered in the next stage. Therefore, we choose to use the MGDC scheme due to its simplicity. In particular, the MGDC scheme can easily and quickly find a set of disks to cover all tags in $\mathcal{T}$ by checking any pair of tags (i.e., the three rules). Then, we can eliminate redundant disks by iteratively picking a disk with the most tags, which is also easy to implement. One may propose using a more sophisticated scheme to find a more accurate classification of tags, but it will be uneconomical in computation, as stage 2 will provide detailed clustering of tags.*

4.2 Stage 2–Tag Clustering

After getting the result of tag classification from stage 1, we further cluster a subset of tags into groups such that every group of tags are within the maximum communication range of a reader, under the premise that a group contains no more than β tags. In other words, each group of tags can be covered by one reader with the β constraint in theory. To do so, we enhance the *agglomerative hierarchical clustering (AHC)* scheme [27], which clusters tags in a recursive manner according to their positions in the sensing field. In particular, our enhanced AHC scheme has the following three steps:

- **Step 1:** Each tag is initially viewed as a single group.
- **Step 2:** We then seek to merge two groups G_i and G_j such that they have the shortest *inter-group distance* $d(G_i, G_j)$, which is defined by the Euclidean distance between two farthest tags t_a and t_b, where $t_a \in G_i$ and $t_b \in G_j$. However, G_i and G_j can be merged into one single group only if both conditions are satisfied: 1) $d(G_i, G_j) \leq 2r_\alpha$ and 2) $|G_i| + |G_j| \leq \beta$. Here, the first condition means that the tags in both G_i and G_j could be within the maximum communication range of a reader, while the second condition applies the β constraint.

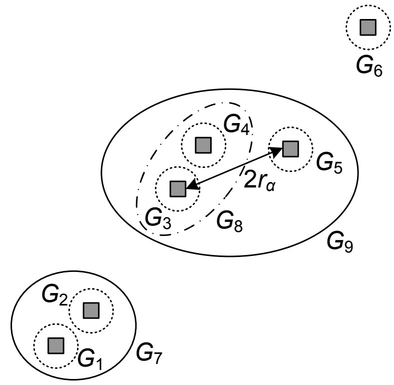


Fig. 5: An example of our enhanced AHC scheme.

- **Step 3:** We repeat step 2 until no groups can be further merged.

Fig. 5 presents an example to illustrate how the enhanced AHC scheme works, where we set $\beta = 4$. In the beginning, we cluster each single tag into one group, so we have groups $G_1 \sim G_6$. Then, we can merge G_1 and G_2 into a new group G_7, and also G_3 and G_4 into a new group G_8. Afterwards, because 1) the inter-group distance $d(G_5, G_8) = d(G_5, G_3) = 2r_\alpha$ and 2) $|G_5| + |G_8| = 3 < \beta$, both G_5 and G_8 can be merged into a new group G_9. Finally, since no clusters can be further merged (due to violation of the two conditions), the enhanced AHC scheme returns three groups G_6, G_7, and G_9 in this example.

Remark 2. *In stage 2, there could exist some schemes to cluster tags in each subset, for example, the popular K-means scheme [28]. We choose to enhance the AHC scheme to cluster tags due to two reasons. First, the AHC scheme clusters tags in a hierarchical manner by putting the tags whose positions are close to each other in the same group. Therefore, it becomes easy to limit the size of each group by the maximum communication range of a reader. Second, the K-means scheme requires the information of the number of groups a prior, but it is difficult to compute the number of readers used to cover a subset of tags in stage 2. On the contrary, the AHC scheme allows us to cluster tags in a bottom-up manner, so it will not encounter the same problem arisen in the K-means scheme.*

4.3 Stage 3–Reader Placement

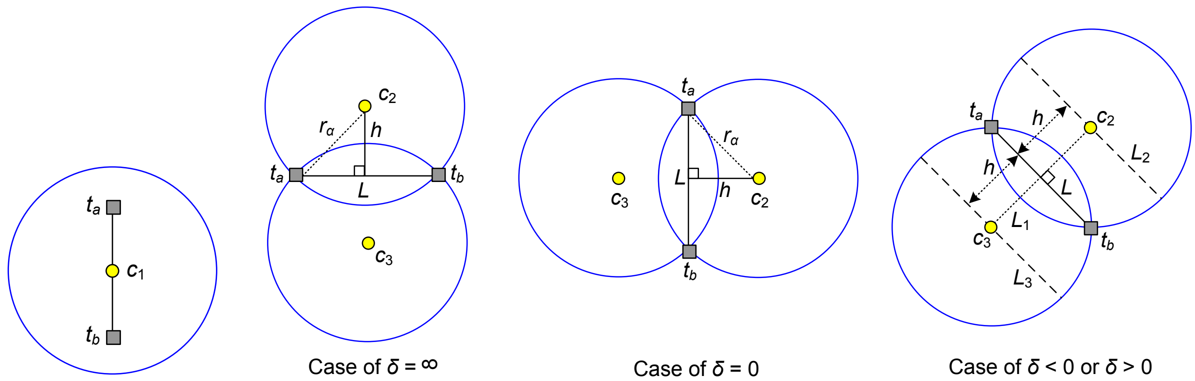
Through clustering tags into groups by the enhanced AHC scheme, we can place one reader to cover each group accordingly. The trivial case is that there exists only one tag in a group. In this case, we directly place one reader at the tag's position and tune the communication range of the reader to r_1 (i.e., the smallest communication range) to save its energy.

For multi-tag case, we can adopt the two farthest tags t_a and t_b in the group to determine the candidate locations to place a reader. Specifically, there are three candidate locations denoted by c_1, c_2, and c_3 in Fig. 6. Let (x_a, y_a) and (x_b, y_b) be the coordinates of t_a and t_b, respectively. Since c_1 is the midpoint of t_a and t_b (as shown in Fig. 6(a)), we can easily calculate its coordinates by

$$c_1 = \left(\frac{x_a + x_b}{2}, \frac{y_a + y_b}{2}\right). \tag{9}$$

Let L be the line passing both tags t_a and t_b. We derive the slope of L as follows:

$$\delta = \frac{y_b - y_a}{x_b - x_a}. \tag{10}$$



(a) (b)

Fig. 6: Three candidate locations to place a reader: (a) c_1 and (b) c_2 and c_3.

The equation of line L is formulated by

$$y = \delta(x - x_a) + y_a. \tag{11}$$

From Fig. 6(b), the candidate locations c_2 and c_3 are in the two opposite sides of L, and we can calculate their coordinates by the three cases below (also indicated by the flow chart in Fig. 2):

- **Case of $\delta = \infty$:** In this case, L is parallel with the x-axis. Because we have $d(c_2, t_a) = d(c_2, t_b) = d(c_3, t_a) = d(c_3, t_b) = r_\alpha$, we can use the Pythagorean theorem to compute the vertical distance from c_2 (also c_3) to L:

$$h = \sqrt{r_\alpha^2 - (d(t_a, t_b)/2)^2}. \tag{12}$$

  Therefore, we can derive the coordinates of c_2 and c_3 as follows:

$$c_2 = \left(\frac{x_a + x_b}{2}, \frac{y_a + y_b}{2} + h\right) \tag{13}$$

$$c_3 = \left(\frac{x_a + x_b}{2}, \frac{y_a + y_b}{2} - h\right) \tag{14}$$

- **Case of $\delta = 0$:** Line L will be perpendicular to the x-axis. Similarly, by obtaining the distance h from Eq. (12), we can calculate the coordinates of c_2 and c_3 as follows:

$$c_2 = \left(\frac{x_a + x_b}{2} + h, \frac{y_a + y_b}{2}\right) \tag{15}$$

$$c_3 = \left(\frac{x_a + x_b}{2} - h, \frac{y_a + y_b}{2}\right) \tag{16}$$

- **Case of $\delta > 0$ or $\delta < 0$:** To find the coordinates of c_2 and c_3, we have to derive the equations of three lines L_1, L_2, and L_3, where L_1 is perpendicular to L which passes both c_2 and c_3, L_2 is parallel with L that passes c_2, and L_3 is parallel with L that passes c_3, as shown in Fig. 6(b). Specifically, the equation of line L_1 is given as follows:

$$y = -\frac{1}{\delta}\left(x - \frac{x_a + x_b}{2}\right) + \frac{y_a + y_b}{2}. \tag{17}$$

  In addition, because L_2 and L_3 have the same slope with L, we can express their equations of lines as follows:

$$L_2: \ y = \delta x + k_2 \tag{18}$$

$$L_3: \ y = \delta x + k_3 \tag{19}$$

  By Eqs. (17) and (18), we can compute the coordinates of c_2:

$$c_2 = \left(\frac{A/\delta + B - k_2}{\delta + 1/\delta}, \frac{A + \delta B + k_2/\delta}{\delta + 1/\delta}\right), \tag{20}$$

  where

$$A = \frac{x_a + x_b}{2} \quad \text{and} \quad B = \frac{y_a + y_b}{2}. \tag{21}$$

  Similarly, we can calculate the coordinates of c_3 through Eqs. (17) and (19):

$$c_3 = \left(\frac{A/\delta + B - k_3}{\delta + 1/\delta}, \frac{A + \delta B + k_3/\delta}{\delta + 1/\delta}\right), \tag{22}$$

  To find the value of k_2 in Eq. (20), we first calculate the vertical distance between two parallel lines L and L_2. In particular, suppose that L_2 locates on the right side of L. From their equations of lines in Eqs. (11) and (18), we can derive their distance by

$$d(L, L_2) = \frac{k_2 - (-\delta x_a + y_a)}{\sqrt{\delta^2 + (-1)^2}}. \tag{23}$$

  In fact, $d(L, L_2)$ will be also equal to h in Eq. (12). Therefore, we can obtain that

$$h = \frac{k_2 - (-\delta x_a + y_a)}{\sqrt{\delta^2 + (-1)^2}}$$
$$\Rightarrow \ k_2 = -\delta x_a + y_a + h\sqrt{\delta^2 + 1}. \tag{24}$$

  On the other hand, we are able to use the vertical distance between two parallel lines L and L_3 to acquire the value of k_3 in Eq. (22):

$$d(L, L_3) = -\frac{k_3 - (-\delta x_a + y_a)}{\sqrt{\delta^2 + (-1)^2}} = h$$
$$\Rightarrow \ k_3 = -\delta x_a + y_a - h\sqrt{\delta^2 + 1}. \tag{25}$$

After identifying all candidate locations, we iteratively select the group that contains the maximum number of uncovered tags. For the group, we place a reader on one of the three candidate locations (i.e., c_1, c_2, and c_3 in Fig. 6) such that 1) all tags in the group can be covered by the reader and 2) the reader has the minimum number of overlapped tags with its conflicting neighbors. In case that these two conditions conflict with each other, the first condition (i.e., covering all tags) always has the highest priority. Fig. 7 presents an example, where seven tags are clustered into two groups. There are two

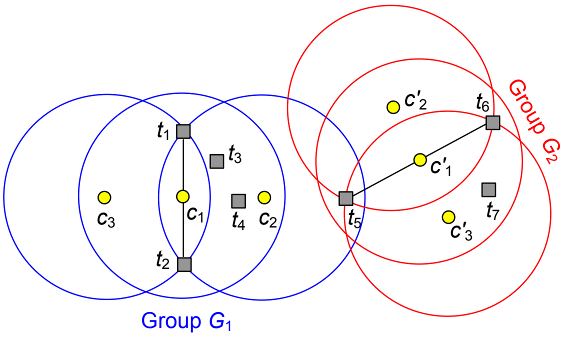


Fig. 7: An example of determining the locations to place readers.

choices c_1 and c_2 that we are able to place a reader to cover all tags in group G_1. However, if we place the reader on c_2, it will cover a tag t_5 in another group (i.e., the overlapped tag). Therefore, we should place the reader on c_1 for group G_1. Then, we can place a reader on either c'_1 or c'_3 to cover all tags in group G_2. In this way, both readers will not conflict with each other.

4.4 Stage 4–Collision Handling

In the last stage, we reorganize the tags covered by conflicting readers in order to satisfy the three constraints of β, γ, and ε. In particular, we select the reader that possesses the maximum number of conflicting neighbors, which is denoted by R_i. If R_i violates any constraint, we adopt two methods to adjust R_i's coverage of tags, which is indicated by the flow chart in Fig. 2.

- **Transferring tags:** This method is invoked when
$$|\mathcal{T}_i^S| + |\mathcal{T}_i^O| \leq \beta, \tag{26}$$
where $\mathcal{T}_i^S$ is the set of tags solely covered by R_i and $\mathcal{T}_i^O$ denotes the set of tags that R_i shares with its conflicting neighbors. Then, we transfer all tags in $\mathcal{T}_i^O$ to R_i and recompute the new location and communication range of each of R_i's conflicting neighbors (including R_i itself) according to the above transfer of tags. Fig. 8(a) illustrates an example, where we set $\beta = 4$ and $\varepsilon = 0$. In the example, reader R_2 has two conflicting neighbors R_1 and R_3, and we have $\mathcal{T}_2^O = \{t_3, t_6\}$. Then, we can transfer all tags in $\mathcal{T}_2^O$ to R_2. In this way, we recalculate the locations of R_1, R_2, and R_3 based on the groups of tags $\{t_1, t_2\}$, $\{t_3, t_4, t_5, t_6\}$, and $\{t_7, t_8\}$ through the scheme proposed in stage 3, respectively. Thus, R_2 will not be conflicting with R_1 and R_3 any longer. In addition, we can also shrink the communication range of R_3 under the premise that it covers both t_7 and t_8, so as to reduce its energy consumption and effect of interference range.
- **Adding readers:** In case that Eq. (26) is violated, it means that there are too many tags in the communication range of R_i. Therefore, we seek to transfer some tags to R_i's conflicting neighbors and adjust R_i's communication range or even add extra reader(s) to cover the residual tags. In particular, this method has the following steps:
  1) Let $\mathcal{T}_{i,j}^O \subseteq \mathcal{T}_i^O$ denote the set of overlapped tags owned by R_i and its conflicting neighbor R_j. If $|\mathcal{T}_j^S| + |\mathcal{T}_j^O| < \beta$ (i.e., R_j still has the quota to cover additional tags), then starting from the tag farthest from R_i, we iteratively transfer each tag in $\mathcal{T}_{i,j}^O$ to R_j, until $|\mathcal{T}_j^S| + |\mathcal{T}_j^O| = \beta$ or $\mathcal{T}_{i,j}^O$ becomes empty. Afterwards, we recompute the location and communication range of R_j by the scheme in stage 3.
  2) After transferring the overlapped tags to conflicting neighbors, we use the scheme in stage 3 again to calculate the location and communication range of R_i to cover the residual tags. If R_i satisfies the constraints of β, γ, and ε, the method finishes. Otherwise, it means that we require more readers to cover these residual tags (denoted by $\mathcal{T}_i^R$).
  3) We then use the K-means scheme to cluster the tags in $\mathcal{T}_i^R$ into groups, whose pseudocode is given in Algorithm 1. In particular, starting from $K = 2$, we cluster these tags and check whether each group of tags can be covered by one reader which satisfies the constraints of β, γ, and ε. If

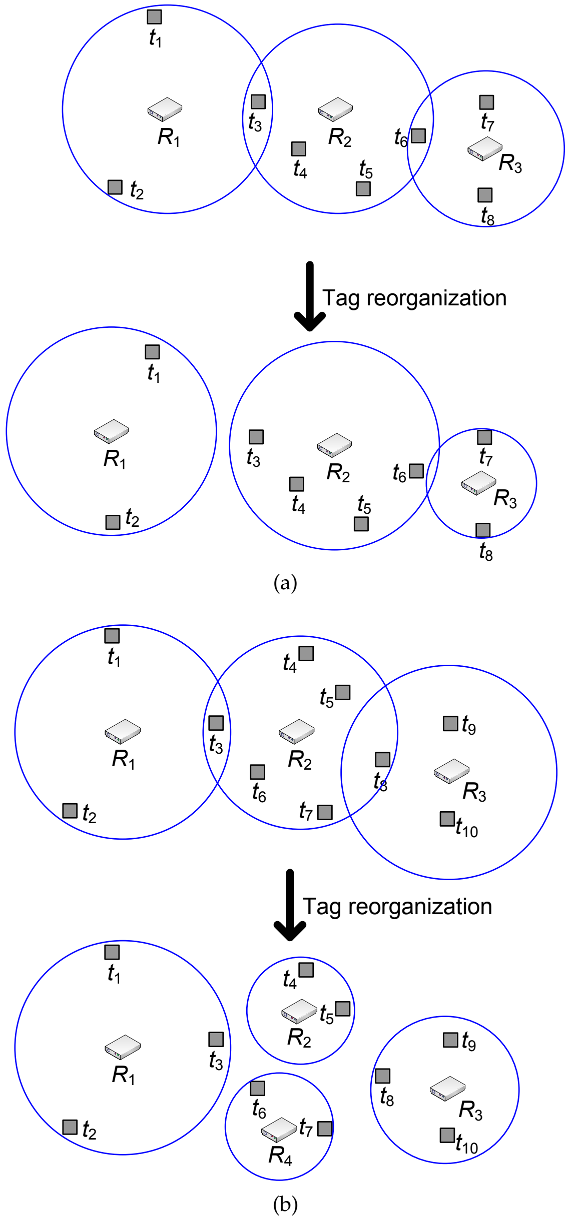


Fig. 8: Two methods used to reorganize tags to deal with the situation of conflicting readers: (a) transferring tags and (b) adding readers.

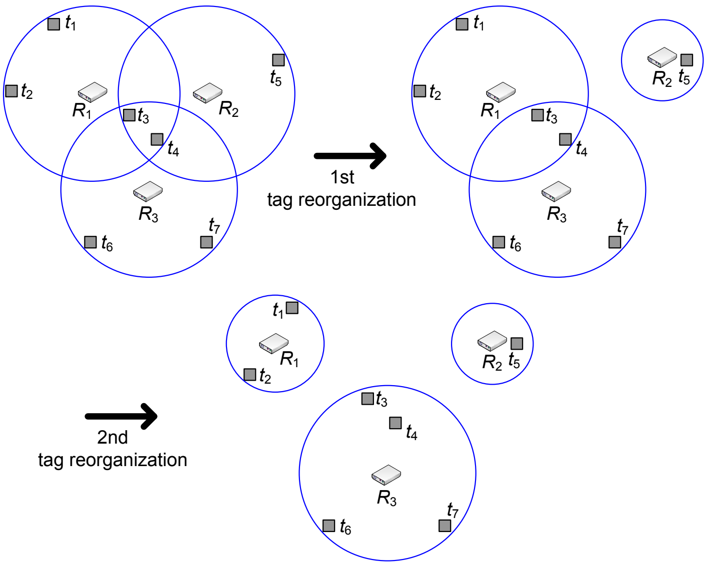


Fig. 9: An example of tag reorganization for the case where three readers cover the same tags.

not, we then iteratively add K by one and repeat step 3, until each reader meets the above three constraints.

Fig. 8(b) gives an example, where we first transfer tags t_3 and t_8 to R_2's conflicting neighbors R_1 and R_3, respectively. Then, we adopt the K-means scheme to cluster the residual tags $\{t_4, t_5, t_6, t_7\}$ into two groups, where each group of tags can be covered by one reader with smaller communication range without conflicting with other readers, as shown in Fig. 8(b).

We iteratively check each reader with conflicting neighbors and solve the problem by the above two methods, until every reader can satisfy the three constraints of β, γ, and ε.

Algorithm 1: K-means scheme

Input: a set $\mathcal{T}_i^R$ of tags and K
Output: Groups $\{G_1, G_2, \cdots, G_K\}$ of tags in $\mathcal{T}_i^R$

```
Randomly select K tags in T_i^R as central points {p_1, p_2, ..., p_K};
repeat
    foreach tag t_j ∈ T_i^R do
        if t_j has the shortest distance to a central point p_k then
            Add t_j to group G_k;
        end
    end
    foreach group G_k do
        Suppose that G_k contains m tags with locations of {(x_1, y_1), (x_2, y_2) ... (x_m, y_m)};
        Find its central point p_k by ( (1/m) Σ_{j=1}^{m} x_j, (1/m) Σ_{j=1}^{m} y_j );
    end
until no groups can be further changed;
```

We remark that when there are three or more readers that cover the same tags, we can execute the schemes of transferring tags or adding tags multiple times to deal with this situation, where in each iteration we consider only two readers. Fig. 9 gives an example, where we set $\beta = 4$ and $\gamma = 1$. In the first iteration, we consider both readers R_1 and R_2. Then, R_2 transfers tags t_3 and t_4 to R_1, so that R_2 can shrink its communication range to cover only tag t_5. However, because R_1 and R_3 still cover both tags t_3 and t_4, which violates the γ constraint. Therefore, R_1 transfers these two tags to R_3 in the second iteration. In this way, all readers are not conflicting with each other.

4.5 Discussion

We then discuss the rationale of our proposed MR2D algorithm. The first two stages seek to cluster tags into groups such that every group contains no more than β tags and it can be covered by just one reader. In particular, stage 1 is an auxiliary operation for stage 2, since we are able to use merely the enhanced AHC scheme in stage 2 to cluster tags. However, using stage 1 helps alleviate the computation overhead of stage 2, because it not only divides tags into independent subsets but also directly handles the simple case where we can deploy one reader to cover a subset of tags. Therefore, stage 2 can employ the *divide-and-conquer* concept by individually dealing with the tags in each small subset. In particular, we tailor the AHC scheme to the need of our tag clustering by adding two conditions that a group will not contain more than β tags and the distance between two farthest tags in a group will not exceed the communication range of a reader.

After clustering tags into different groups, the objective of stage 3 is to find the suitable location(s) to place a reader to cover the tags in each group. One may suggest using a more sophisticated scheme to compute every possible location to place the reader, for example, finding each disk whose circumference passes any two or three tags in the group. Nevertheless, such a scheme incurs very high computation complexity. To conquer this problem, we adopt only the two farthest tags in a group to identify three candidate locations to place the reader, which significantly reduces the computation overhead. Afterwards, we can use both the transferring-tags and adding-readers schemes in stage 4 to deal with the case where a reader violates the three constraints defined in Section 3 due to its conflicting neighbors and overlapped tags. In this way, we are able to deploy fewer readers and shrink their communication range to save energy.

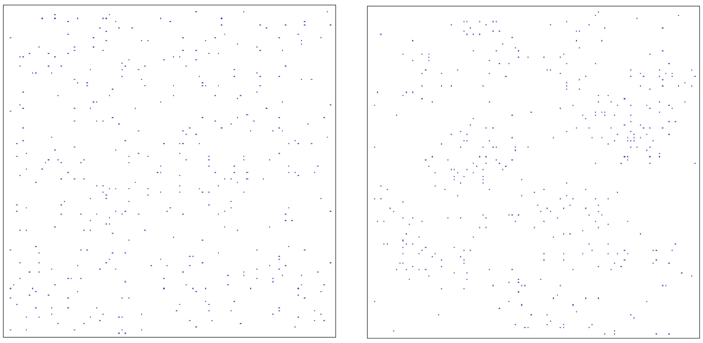
We remark that our MR2D algorithm operates based on the assumption of a binary communication model. In some situations, however, the probability that a reader can successfully detect a tag will decay with the distance from the reader to the tag. For instance, given the communication range r_k, the detection probability of a tag t_j by a reader R_i can be calculated as follows [29]:

$$Prob(R_i, t_j) = \begin{cases} e^{-\psi d(R_i, t_j)} & \text{if } d(R_i, t_j) \le r_k \\ 0 & \text{otherwise,} \end{cases} \tag{27}$$

where ψ is a parameter that represents the physical characteristic of the reader. If we want to make sure that every tag within the communication range of R_i has a detection probability no smaller than a given threshold p_{th}, we can compute a *virtual* communication range r'_k by

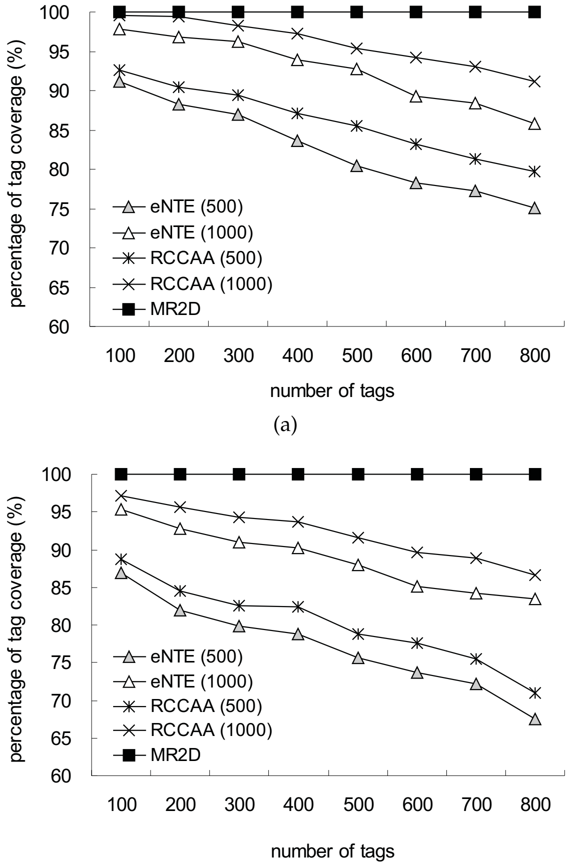
$$e^{-\psi r'_k} = p_{\text{th}} \Rightarrow r'_k = -\frac{\ln p_{\text{th}}}{\psi}. \tag{28}$$

By replacing r_k by r'_k, it is guaranteed that every tag in R_i's communication range will have a detection probability of at



(a) (b)

Fig. 10: Two distributions of tags in the sensing field: (a) random distribution and (b) congregating distribution.



(a)

(b)

Fig. 11: Comparison on the percentage of tag coverage: (a) random distribution and (b) congregating distribution.

least $p_{\rm th}$. In this way, our MR2D algorithm can be also applied to the probabilistic communication model.

5 Simulation Study

We develop a simulator in Java to evaluate the performance of our proposed MR2D algorithm. The sensing field is modeled by a 200 m×200 m square, inside which there are 100 to 800 tags. We consider two distributions of tags in the sensing field, as shown in Fig. 10:

- **Random distribution:** All tags are arbitrarily deployed in the sensing field by following the uniform distribution.
- **Congregating distribution:** We first adopt 10% of tags as *seeds* to be deployed in the sensing field. Then, 70% of tags are deployed in the surrounding regions of these seed tags. Afterwards, the remaining 20% of tags are also arbitrarily deployed.

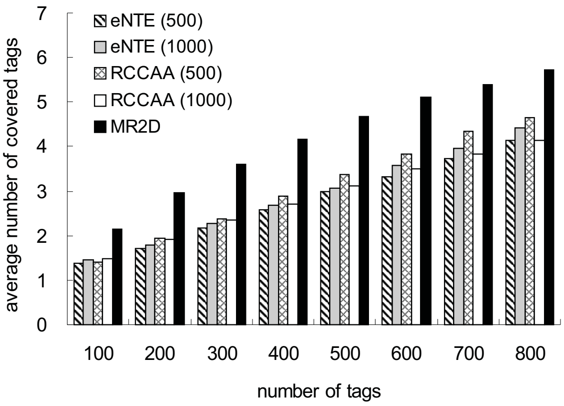
Each reader has three communication radiuses of 1 m, 5 m, and 10 m. A reader is allowed to cover at most ten tags (i.e., $\beta = 10$). In addition, we set $\gamma = 2$ and $\varepsilon = 1$, so any two readers can share no more than two tags and each reader has zero or one conflicting neighbor. We compare our MR2D algorithm with two reader deployment methods, NTE and RCCAA, discussed in Section 2. However, the original NET method does not consider tag and reader collision. Therefore, we develop an extended version of NET, called *eNET*. The eNET method adopts NET to select readers, where the communication radius of each reader is set to 10 m in the beginning. Then, we check if every reader obeys the β constraint. If a reader covers more than β tags, we shrink its communication range step by step until it meets the β constraint. Then, we employ the schemes of transferring tags and adding tags in Section 4.4 to make two conflicting readers satisfy the γ and ε constraints[3]. On the other hand, the RCCAA method adopts the β constraint but does not allow a reader to have conflicting neighbors (i.e., $\gamma = 0$ and $\varepsilon = 0$). Both the eNTE and RCCAA methods assume that there have been readers placed in the sensing field and seek to turn off unnecessary ones. Therefore, we also arbitrarily place 500 and 1000 readers (by using the uniform distribution) in the sensing field for the eNTE and RCCAA methods. Below, we measure tag coverage, the number of readers, and energy consumption by the eNTE, RCCAA, and MR2D methods. Afterwards, we investigate the effect of parameters β and γ on our MR2D algorithm.

3. When the two schemes compute new locations of readers, the eNTE method will activate the readers which are closest to these locations. In case that no such readers can be found, we turn off the conflicting reader that covers the fewest tags to satisfy the γ and ε constraints.

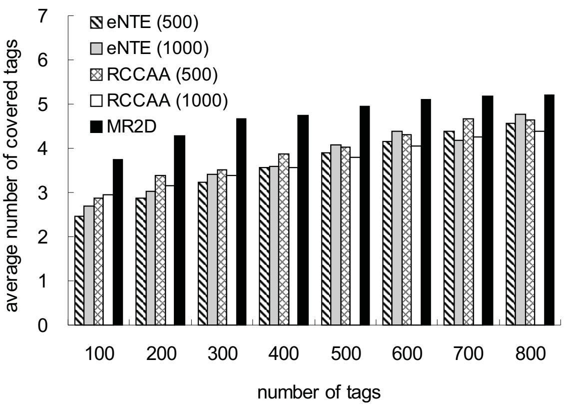
Remark 3. *The eNTE method provides another solution to the MR2D problem, as it replaces the first three stages of our MR2D algorithm by a greedy approach (i.e., finding readers with larger weights) but still uses stage 4 to deal with the constraints of β, γ, and ε. We can use the eNTE method to observe the effect of the first three phases of the MR2D algorithm on system performance. On the other hand, the RCCAA problem is similar to the MR2D problem, except that it considers only the β constraint. In [21], Liu et al. prove that the RCCAA method has an approximation ratio of θ, where θ is the maximum degree in the given graph that models the RFID system. Therefore, we also compare our MR2D algorithm with the RCCAA method to observe how γ and ε constraints affect system performance.*

5.1 Tag Coverage

In the first experiment, we measure the percentage of tag coverage by different methods, as illustrated in Fig. 11. Both the eNTE and RCCAA methods seek to select readers from arbitrarily placed ones to cover the maximum number of tags. Consequently, they result in higher tag coverage with 1000 readers, as there are more choices of readers. For the eNTE method, it greedily selects readers that cover more tags and have more neighbors (due to the NET's design). However, when the β constraint is violated, readers will shrink their communication range and give up a part of tags. In addition,

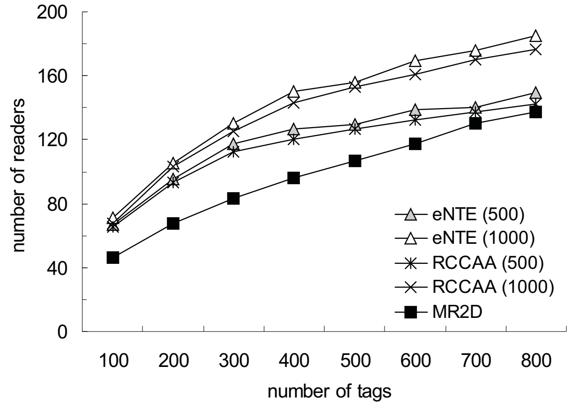


(a)

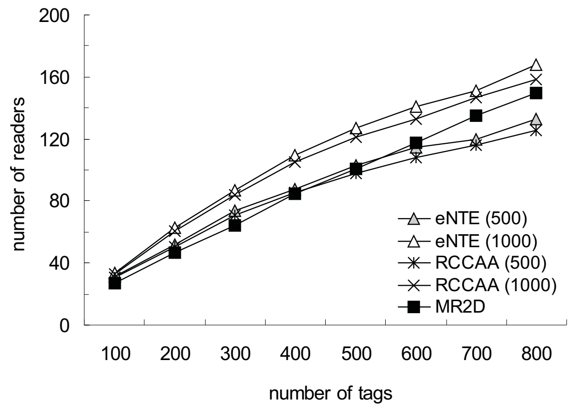


(b)

Fig. 12: Comparison on the average number of tags covered by each reader: (a) random distribution and (b) congregating distribution.



(a)



(b)

Fig. 13: Comparison on the number of readers deployed: (a) random distribution and (b) congregating distribution.

when the eNET method cannot find suitable readers to deal with the γ and ε constraints, some of selected (conflicting) readers have to be deactivated. That is why the eNTE method results in the lowest percentage of tag coverage, especially when there are more tags. On the other hand, the RCCAA method does not allow readers to have conflicting neighbors (i.e., $\gamma = \varepsilon = 0$). Therefore, its tag coverage significantly degrades when the number of tags grows. Such a phenomenon is more obvious under the congregating distribution of tags, because some regions could have pretty large density of tags. In this case, it becomes more difficult for the RCCAA method to select readers to cover these regions. In contrast with both the eNTE and RCCAA methods, our MR2D algorithm always achieves 100% of tag coverage, since it can efficiently cluster tags according to their positions and compute the suitable locations of readers to cover each groups of tags. Moreover, by applying the γ and ε constraints, our MR2D algorithm can more flexibly handle the case with high density of tags, as comparing with the RCCAA method.

Next, we study the average number of tags covered by each reader, as presented in Fig. 12. On the whole, both the eNTE and RCCAA methods allow a reader to cover more tags under the congregating distribution of tags, because they can find out relatively fewer readers to cover the regions where most tags aggregate. In our MR2D algorithm, each reader can cover more tags than both the eNTE and RCCAA methods, especially under the random distribution of tags. In particular, the MR2D algorithm adaptively adjusts the communication range of a reader or even adds extra readers to cover the overlapped tags when the γ and ε constraints are violated. Under the congregating distribution of tags, the possibility of violation of the γ and ε constraints significantly increases, since tags are inclined to aggregate in some regions. That is why the average number of covered tags decreases in the MR2D algorithm under the congregating distribution of tags. In sum, this experiment exhibits that our MR2D algorithm can provide complete coverage of tags and allow each reader to cover more tags (with the consideration of the β constraint), as comparing with the eNTE and RCCAA methods.

5.2 Number of Readers

The second experiment evaluates the number of readers deployed by the eNTE, RCCAA, and MR2D methods. Fig. 13(a) presents the simulation result under the random distribution of tags. The eNTE and RCCAA methods use fewer readers in the case of 500 initial readers than that in the case of 1000 initial readers. The reason can be found in Fig. 11(a), as both methods have lower percentages of tag coverage in the case of 500 initial readers (in particular, 75.1%$\sim$91.2% tags are covered by the eNTE method while 79.8%$\sim$92.6% tags are covered by the RCCAA method). Comparing with these two methods, our MR2D algorithm can significantly save the number of readers, because it can precisely calculate the locations of readers according to the positions of tags, with the consideration of β, γ, and ε constraints.

Fig. 13(b) gives the simulation result under the congregating distribution of tags. Since most tags will aggregate within some regions, it becomes easier to find a reader to cover more

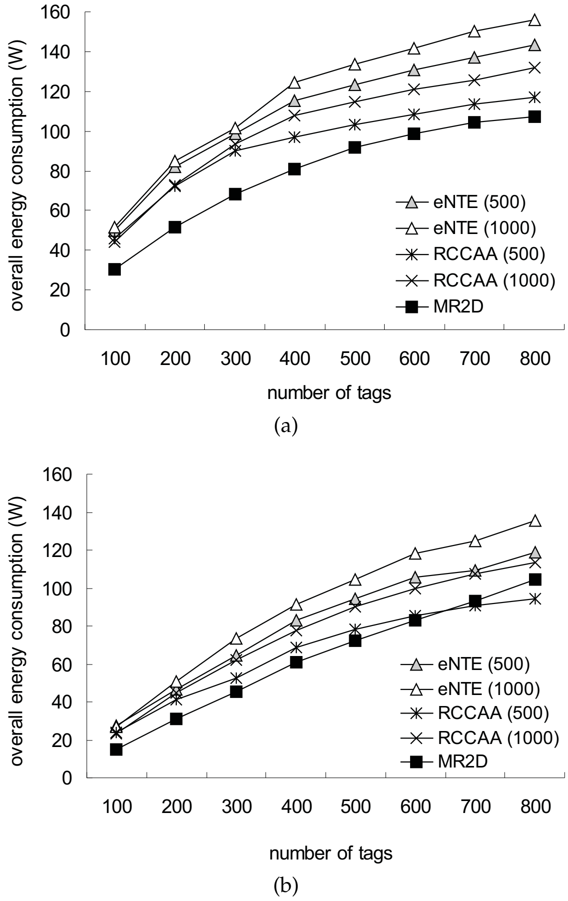


Fig. 14: Comparison on the amount of energy consumed by total readers: (a) random distribution and (b) congregating distribution.

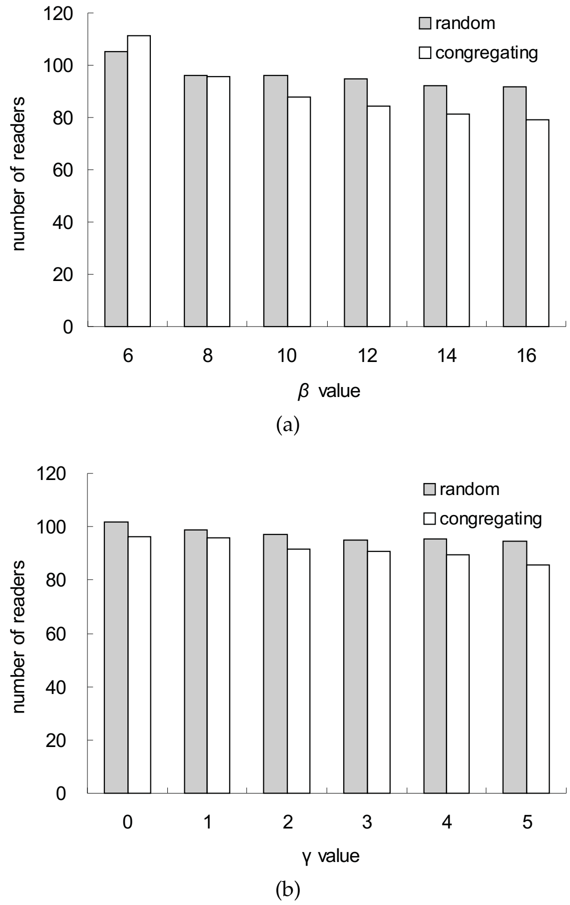


Fig. 15: Evaluation of the effect of parameters on the MR2D algorithm: (a) β value and (b) γ value.

tags. Therefore, all methods require relatively fewer readers to cover tags. With the similar reason from Fig. 11(b), both the eNTE and RCCAA methods with 500 initial readers deploy fewer readers than our MR2D algorithm when the number of tags exceeds 500, because they provide only partial coverage of tags (in particular, 67.5%$\sim$86.9% tags are covered by the eNTE method while 70.9%$\sim$88.7% tags are covered by the RCCAA method). To satisfy the γ and ε constraints, our MR2D algorithm would add additional readers to cover those overlapped tags in its stage 4, so it requires more readers accordingly. In general, the MR2D algorithm can use fewer readers to cover all tags as compared with other methods.

5.3 Energy Consumption

We also measure the amount of energy spent by the deployed readers in Fig. 14. Generally speaking, all methods make readers consume less energy under the congregating distribution of tags, because they employ fewer readers to cover tags. In the eNTE method, each reader keeps the maximum radius for communication if it does not violate any constraint of β, γ, and ε. On the other hand, the RCCAA method considers adjusting the communication range of readers to cover tags. Thus, the eNTE method consumes more energy than the RCCAA method. Our MR2D algorithm can adaptively shrink the communication range of a reader based on the positions of tags that the reader covers. Therefore, it can further reduce the energy consumption of readers comparing with the RCCAA method. Notice that the RCCAA method (with 500 initial readers) deploys fewer readers than the MR2D algorithm (referring to Fig. 13(b)) due to partial coverage of tags, so it will have less energy consumption than the MR2D algorithm under the congregating distribution of tags, when there are more than 600 tags in the sensing field. From this experiment, we demonstrate that our MR2D algorithm can efficiently save the energy of readers to query their tags by properly adjusting their communication range.

5.4 Effect of Parameters

In the last experiment, we study the effect of parameters β and γ on the MR2D algorithm, where there are 400 tags in the sensing field and we set $\varepsilon = 1$. Fig. 15(a) shows the effect of β, where $\gamma = 2$. We can observe that the number of readers significantly decreases when the value of β grows, especially under the congregating distribution of tags. It is because a reader is allowed to cover more tags as β increases. On the other hand, Fig. 15(b) presents the effect of γ, where $\beta = 10$. We can observe that the effect of γ becomes slight under the random distribution of tags. In such distribution, the possibility that two readers share more overlapped tags will decrease. Consequently, increasing the γ value may not help further save the number of readers.

6 Conclusion and Future Work

In an RFID system, how to deploy readers significantly affects both cost and performance of the system. Many methods seek to select a subset of readers from arbitrarily placed ones to cover the maximum number of tags, whose results depend on the initial placement of readers. Consequently, this paper proposes the MR2D problem to deploy the minimum number of readers and determine their communication range to provide complete coverage of tags, under the consideration of reader and tag collision. Our solution involves in not only efficient clustering of tags but also reader arrangement for clusters. Moreover, the communication range of each reader is adaptively adjusted to save its energy and satisfy the constraints related to reader and tag collision. By conducting simulations on two distributions of tags, we demonstrate that our proposed MR2D algorithm not only provides complete coverage of tags by using fewer readers but also saves more energy of these readers, as comparing with the eNTE and RCCAA methods.

In this paper, we aim at finding the smallest number of readers required to cover tags in a centralized manner. Another thought is to have readers as mobile agents (or robots) and develop a decentralized algorithm to make these readers move to cover tags on their own. At first glance, the problem may be similar to point coverage by mobile sensors [30] or flock formation by mobile robots [31], [32]. However, we should consider the β, γ, and ε constraints, and also how to adjust the communication range of different readers. These issues pose challenges and deserve further investigation for the future work.

REFERENCES

[1] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Comm. Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] D.K. Klair, K.W. Chin, and R. Raad, "A survey and tutorial of RFID anti-collision protocols," *IEEE Comm. Surveys & Tutorials*, vol. 12, no. 3, pp. 400–421, 2010.

[4] D.W. Engels and S.E. Sarma, "The reader collision problem," *Proc. IEEE Int'l Conf. Systems, Man and Cybernetics*, 2002, pp. 1–6.

[5] Y. Bai, X.W. Xuan, J.F. Teng, and L.Y. Zhang, "A RFID anti-collision algorithm based on distributed power control," *Proc. IEEE Int'l Conf. Wireless Comm. Networking and Mobile Computing*, 2010, pp. 1–4.

[6] J. Waldrop, D.W. Engels, and S.E. Sarma, "Colorwave: an anticollision algorithm for the reader collision problem," *Proc. IEEE Int'l Conf. Comm.*, 2003, pp. 1206–1210.

[7] A. Jedda and H.T. Mouftah, "Decentralized RFID coverage algorithms with applications for the reader collisions avoidance problem," *IEEE Trans. Emerging Topics in Computing*, vol. 4, no. 4, pp. 502–515, 2016.

[8] V. Chawla and D.S. Ha, "An overview of passive RFID," *IEEE Comm. Magazine*, vol. 45, no. 9, pp. 11–17, 2007.

[9] A.E. Al-Fagih, F.M. Al-Turjman, H.S. Hassanein, and W.M. Alsalih, "Coverage-based placement in RFID networks: an overview," *Proc. Int'l Conf. Mobile, Ubiquitous, and Intelligent Computing*, 2012, pp. 220–224.

[10] A. Oztekin, F. Mahdavi, K. Erande, Z. Kong, L.K. Swim, and S.T.S. Bukkapatnam, "Criticality index analysis based optimal RFID reader placement models for asset tracking," *Int'l Journal of Production Research*, vol. 48, no. 9, pp. 2679–2698, 2010.

[11] W. Yoon and N.H. Vaidya, "RFID reader collision problem: performance analysis and medium access," *Wireless Comm. and Mobile Computing*, vol. 12, no. 5, pp. 420–430, 2012.

[12] A. Suriya and J.D. Porter, "Genetic algorithm based approach for RFID network planning," *Proc. IEEE Region 10 Conf.*, 2014, pp. 1–5.

[13] Y.C. Wang, C.C. Hu, and Y.C. Tseng, "Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks," *Proc. IEEE Wireless Internet Conf.*, 2005, pp. 114–121.

[14] B. Carbunar, M.K. Ramanathan, M. Koyuturk, C. Hoffmann, and A. Grama, "Redundant reader elimination in RFID systems," *Proc. IEEE Conf. Sensor and Ad Hoc Comm. and Networks*, 2005, pp. 176–184.

[15] C.H. Hsu, Y.M. Chen, and C.T. Yang, "A layered optimization approach for redundant reader elimination in wireless RFID networks," *Proc. IEEE Asia-Pacific Service Computing Conf.*, 2007, pp. 138–145.

[16] K. Ali, H.S. Hassanein, and W. Alsalih, "Using neighbor and tag estimations for redundant reader eliminations in RFID networks," *Proc. IEEE Wireless Comm. and Networking Conf.*, 2011, pp. 832–837.

[17] A. Jedda, M.G. Khair, and H.T. Mouftah, "Distributed algorithms for the RFID coverage problem," *Proc. IEEE Int'l Conf. Comm.*, 2013, pp. 1758–1762.

[18] N. Rashid, S. Choudhury, and K. Salomaa, "CARRE: cellular automaton based redundant readers elimination in RFID networks," *Proc. IEEE Int'l Conf. Comm.*, 2016, pp. 1–6.

[19] H. Gutowitz, *Cellular Automata: Theory and Experiment*. Cambridge: The MIT Press, 1991.

[20] S. Tang, C. Wang, X.Y. Li, and C. Jiang, "Reader activation scheduling in multi-reader RFID systems: a study of general case," *Proc. IEEE Int'l Parallel & Distributed Processing Symp.*, 2011, pp. 1147–1155.

[21] B.H. Liu, N.T. Nguyen, V.T. Pham, and Y.H. Yeh, "A maximum-weight-independent-set-based algorithm for reader-coverage collision avoidance arrangement in RFID networks," *IEEE Sensors Journal*, vol. 16, no. 5, pp. 1342–1350, 2016.

[22] S. Basagni, "Finding a maximal weighted independent set in wireless networks," *Telecomm. Systems*, vol. 18, no. 1-3, pp. 155–168, 2001.

[23] M. Bouet and A.L. dos Santos, "RFID tags: positioning principles and localization techniques," *Proc. IFIP Wireless Days*, 2008, pp. 1–5.

[24] L. Yang, Y. Qi, J. Han, C. Wang, and Y. Liu, "Shelving interference and joint identification in large-scale RFID systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 26, no. 11, pp. 3149–3159, 2015.

[25] GAO RFID Inc. [Online]. Available: http://gaorfid.com/

[26] Y.C. Wang, Y.F. Chen, and Y.C. Tseng, "Using rotatable and directional (R&D) sensors to achieve temporal coverage of objects and its surveillance application," *IEEE Trans. Mobile Computing*, vol. 11, no. 8, pp. 1358–1371, 2012.

[27] A. Fernandez and S. Gomez, "Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms," *Journal of Classification*, vol. 25, no. 1, pp. 43–65, 2008.

[28] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. Cambridge: Academic Press, 2001.

[29] Y.C. Wang, C.C. Hu, and Y.C. Tseng, "Efficient placement and dispatch of sensors in a wireless sensor network," *IEEE Trans. Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.

[30] Y.C. Wang, "Mobile sensor networks: system hardware and dispatch software," *ACM Computing Surveys*, vol. 47, no. 1, pp. 12:1–12:36, 2014.

[31] R. Vidal, O. Shakernia, and S. Sastry, "Following the flock," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 14–20, 2004.

[32] L.A.V. Reyes and H.G. Tanner, "Flocking, formation control, and path following for a group of mobile robots," *IEEE Trans. Control Systems Technology*, vol. 23, no. 4, pp. 1268–1282, 2015.