Distributed Algorithms for RFID Readers Channel Assignment: Performance and Fault Tolerance

John Sum [1], *Kevin Ho* [2]
[1]Institute of Technology Management, National Chung Hsing University
Taichung, Taiwan ROC. Email: pfsum@nchu.edu.tw.
[2]Department of Computer Science and Communication Engineering
Providence University, Sha-Lu
Taichung, Taiwan ROC. Email: ho@pu.edu.tw.

Abstract

RFID readers channel assignment is a key problem to succeed in RFID system and thus various distribute algorithms were proposed in the past few years. Some of these algorithms like Distributed Color Selection (DCS) algorithm, working for fixed number of channels, are categorized as non-progressive approach. While others like Colorwave, determining the number progressively, are categorized as progressive approach. In this paper, a comparative analysis on all these distributed algorithms is elucidated and their performance in the senses of their convergence, channel distribution and fault tolerance are presented. First, it is found that DCS fails to assign collision-free channels for random reader network. Second, for the non-progressive approach, heuristic and simulated annealing (SA)-based algorithms can produce collision-free solutions. In term of the number of channels required, heuristic algorithm will need slightly more number than the SA-based algorithms. In term of the convergence, SA-based algorithms need much longer time to obtain the solutions. Third, for the progressive approach, heuristic algorithm, SA-based algorithms and Colorwave are able to determine the number automatically and simultaneously generate collision-free solution. However, SA-based algorithms require much longer time than the heuristic algorithm and Colorwave. Colorwave requires more channels than the others. Forth, SA-based algorithms can generate nearly even distributed channel assignments. Finally, we derive analytically the expected number of collided readers if some readers in the network could flip randomly to other channel. We further show that the channel assignments obtained by SA-based algorithms are of least fault tolerance.

Keywords: Assignment, RFID, Simulated Annealing, Simulation, Supply chain management

1. Introduction

Applications of RFID technologies have been succeeded in various industrial sectors, particularly in the retail business and medical & health industry [16, 19, 22, 23]. In recent years, adoption and implementation of RFID systems have also indicated a dramatic growth and

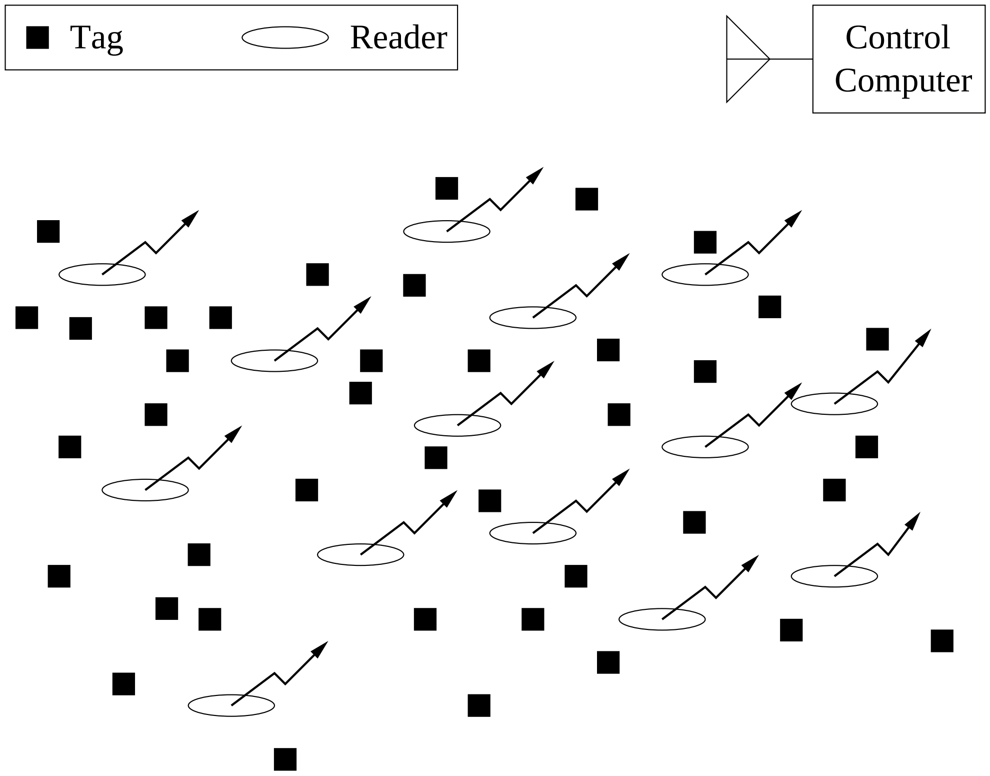


Figure 1: An RFID reader network, with readers communicating to the control computer via a wireless LAN.

the growth will be continue in subsequent years [8, 22]. Scope of applications have also been expanded from inventory control [1] to social networking for dogs [4].

A general design of RFID system with unstructured locations of readers is shown in Figure 1. It consists of three types of components, (a) RFID readers, (b) RFID tags and (c) a control computer. Readers are deployed in an unstructured locations and each has a unique id. Readers communicate with the control computer over usual wireless LAN protocol IEEE 802.11b or 802.11g [15], and pass the data to it for storage. An RFID tag is attached to a single object. Again, each tag has its own id for identification and specific information, such as product code and customer id. Each reader talks with the tags within its vicinity through a specific communication protocol [2, 6, 7]. The control computer is served as a centralized server for the readers. On one hand, it coordinates the interrogations between the readers and the tags so as to avoid interferences (i.e. collisions) between one and other. On the other hand, it receives data transmitted from the readers, consolidates and then deposits in a database.

Since the communication amongst readers and tags rely on radio transmission, each tag (or reader) can simultaneously receive multiple signals from different readers. Interference can easily occur amongst readers and tags if two of more readers use the same frequency band for interrogation and their locations are within the transmission range of one another. These two phenomena are usually called the reader-to-tag collision and reader-to-reader collision [5, 17].

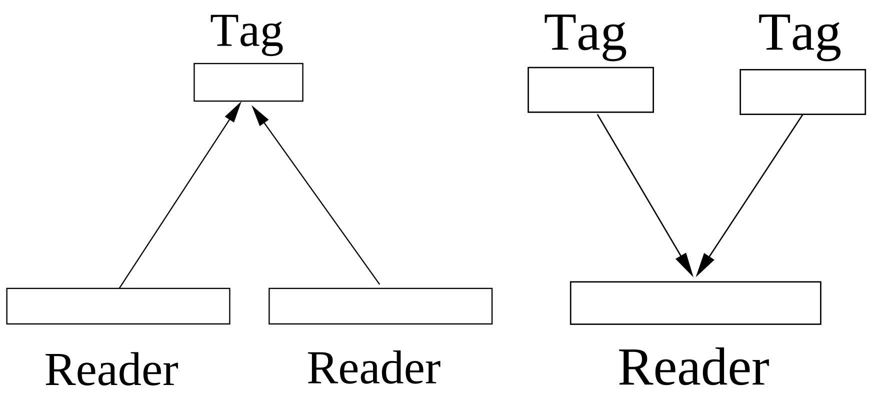


Figure 2: Tag collision problems.

Reader-to-tag collision occurs when two readers simultaneously interrogate a tag, Figure 2. The signals from both readers interfere each others. The tag cannot receive anything. Reader-to-reader collision occurs when one reader receives signal from its neighbor reader of same frequency, Figure 3. The problem of this reader-to-reader collision is called reader collision problem. As data transmission can largely be hampered by collision, many researches were conducted in the past few years to overcome such effect [3, 9, 10, 12, 13, 20, 21, 24]. One should note that channel assignment problem is equivalent to the Graph Coloring Problem, meaning that the problem is NP completed. No polynomial time algorithm exists for the solution [14].

While various distributed algorithms have been developed for readers channel assignment, a number of practical problems are yet to be solved.

1. What is the number of channels required for a collision-free channel assignment ?
2. What are the computational complexity of these algorithms ?
3. Are these algorithms applicable to dense reader networks with unstructured topology ?
4. What are the utilization rates of the assignments if readers have random faults ?

To solve these problems, we present in this paper a comparative analysis on the distributed algorithms that are applied to readers channel assignment. For ease of presentation, we categorize these algorithm as non-progressive and progressive approaches. For non-progressive approach, the algorithms are applied with a pre-defined number of available channels. For the non-progressive approach, this number is not pre-defined. The algorithms search for it automatically. Their performance in the senses of their convergence, channel distribution and fault tolerance are elucidated.

The rest of the paper is organized as follows. The reader collision problem will be described in the next section. The distributed algorithms for readers channel assignment is elucidated in Section 3. Section 4 presents the simulation results on performance of the algorithms in a dense reader networks consisting of 250 readers and their average node degree is 15. The expected number of collided readers, if some readers in the network could flip randomly to

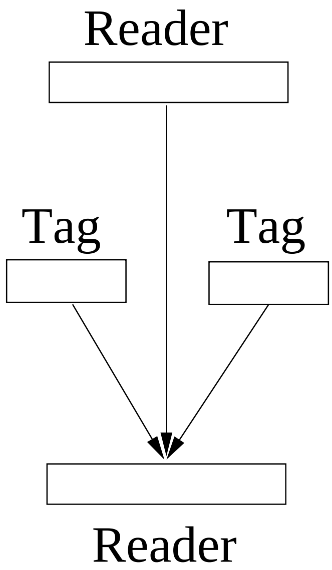


Figure 3: Reader collision problem.

other channel, is derived in Section 5. The fault tolerance the channel assignments obtained by those algorithms are then analyzed. Finally, Section 6 gives a conclusion of the paper.

2. Channel Assignment

Channel assignment problem is a key problem in RFID reader network management, in which neighboring readers are assigned with different channels. So that, collision amongst readers can be avoided. To solve the collision-free channel assignment problem, the following assumptions are made. (1) A pair of readers will have collision if their distance apart is within a range r and they are collecting data in the same channel. (2) Readers are not able to select their frequency bands. (3) The readers are deployed uniformly random within an area of $100m \times 100m$. Their locations are fixed once they have been deployed. (4) Each reader can only assigned with one channel in a cycle for interrogation. (5) No mobile reader is allowed within the area of deployment.

Besides, we suppose that the channel assignment algorithm is solved by the control computer. Once the solution has obtained, the control computer will send message informing the readers the channel being assigned. The readers will thus record their channels being assigned and wait for the synchronization signal from the control computer. Once the signal has been received, each reader will then operate to interrogate at the dedicated channel and reading the tags' data. Whenever its interrogation has been finished, the reader will send data back to the control computer in the consecutive channel. Communication between the control computer and the RFID readers are implemented by wireless LAN.

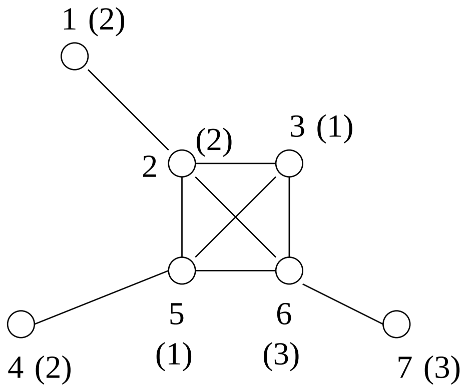


Figure 4: A simple reader network. The numbers inside the parenthesis correspond to the channels being assigned.

2.1 Formal definition of collision

Mathematically, a reader network of N readers can be represented by a graph $\mathcal{G} = (V, E)$. Here V the index set of readers, i.e. $V = \{1, 2, \cdots, N\}$ and the channels assignment is denoted by τ, where $\tau = (\tau_1, \tau_2, \cdots, \tau_N)$. $E = (e_{ij})_{N\times N} \in \{0,1\}^{N\times N}$ is a binary matrix corresponding to the neighborhood relation, i.e.

$$e_{ij} = \begin{cases} 1 & \text{if } d(i,j) \le r \\ 0 & \text{if } d(i,j) > r. \end{cases} \tag{2.1}$$

for all $i, j = 1, 2, \cdots, N$ and $i \neq j$. Here $d(i,j)$ is the distance between the reader i and the reader j. Besides, a reader cannot be a neighbor of itself, i.e. $e_{ij} = 0$. Let τ_i (for $i = 1, 2, \cdots, N$) be the channel being allocated for the i^{th} reader. The collision matrix $C = (c_{ij})_{N\times N} \in \{0,1\}^{N\times N}$ can then be defined. For $i, j = 1, 2, \cdots, N$ and $i \neq j$,

$$c_{ij}(\tau_i, \tau_j, e_{ij}) = \begin{cases} 1 & \text{if } \{e_{ij} = 1\} \text{ and } \{\tau_i = \tau_j\} \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

For illustration, Figure 4 shows a simple reader network comprising seven readers. We assume that there are three channels available. The matrix E and C are given by

$$E = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \text{ and } C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

2.2 Measure of reader collisions

With this definition, the number of collision pairs (CP) in a reader network can be defined as follows :

$$CP(\tau, E) = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}(\tau_i, \tau_j, e_{ij}). \tag{2.3}$$

Which is a measure to the quality of a channels allocation and a reader collision problem can hence be stated. For a RFID reader network $\mathcal{G}$ of N readers and the maximum number of channels in a cycle is T, find an allocation τ such that $CP = 0$. Clearly, this problem is equivalent to the Graph Coloring Problem that is NP hard [14].

3. Distributed Channel Assignment Algorithms

Various algorithms can be applied to solve this reader collision problem. In this paper, we focus on the distributed algorithms in which the algorithm can be conducted distributively by all the readers. Each reader can update its channel only based on the information of its neighbor. No central computer is needed.

For these distributed algorithms, we also categorized them into non-progressive approach and progressive approach. Non-progressive type algorithms deal with the condition that the available channel number is fixed. While progressive type algorithms deal with unlimited number of channels.

3.1 Non-progressive approach for fixed number of channels

For fixed number of channels, five algorithms could be applied. One is the heuristic algorithm. Three are simulated annealing algorithms and one is the Distributed Color Selection.

3.1.1 Heuristic algorithm

For the heuristic approach, a reader is randomly selected in each iteration and is assigned with a channel that can minimize the number of collision pairs. Let say, the i^{th} reader has been selected, the channel τ_i^* to be assigned satisfies the following heuristic rule.

$$\sum_{j=1}^{N} c_{ij}(\tau_i^*, \tau_j, e_{ij}) \leq \sum_{j=1}^{N} c_{ij}(\tau_i, \tau_j, e_{ij}), \tag{3.1}$$

for all $\tau_i \neq \tau_i^*$. The steps can be summarized in Figure 5.

3.1.2 Simulated annealing algorithms

For the simulated annealing approach, the steps are similar except that the change of new channel is not deterministic. Let Δ be the change of total number of collision pairs in the k^{th}

Heuristic Algorithm

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random channels allocation.

S2 Random select a reader, say i.

S3 If reader's channel assignment has no collision to its neighbor readers, then goto S2.

S4 If reader's channel assignment has collision to its neighbors, select a new $\tau_i^* \in \{1, 2, \cdots, T\}$ such that the number of collision pairs between the reader and its neighbors is the minimum.

S5 Repeat steps S2 to S4 until no more improvement can be made.

Figure 5: Heuristic Algorithm

iteration and the i^{th} reader has been selected.

$$\Delta = \sum_{j=1}^{N} c_{ij}(\tau_i^*, \tau_j, e_{ij}) - \sum_{j=1}^{N} c_{ij}(\tau_i, \tau_j, e_{ij}). \tag{3.2}$$

The transition probability of the change of a reader's channel from τ_i to τ_i^* is given by

$$P(\text{New channel} = \tau_i^*) = \begin{cases} 1 & \text{if } \Delta \le 0 \\ \exp\left\{-\frac{\Delta}{\lambda(k)}\right\} & \text{if } \Delta > 0. \end{cases} \tag{3.3}$$

$\lambda(k)$ is the cooling temperature and it decreases as k increases, see Step S6 in Figure 6. Simulated annealing can avoid the search being stuck in a local minimum.

Three common cooling schemes are used.

Constant Temperature For all $k \ge 1$ and $a \ll 1$ is a small constant,

$$\lambda(k) = a. \tag{3.5}$$

Geman-Geman Rule For all $k \ge 1$ and $b > 0$,

$$\lambda(k) = \frac{b}{\log(k+1)}. \tag{3.6}$$

Kirkpatrick *et al* Rule For all $k \ge 1$, $0 < \alpha < 1$ and $c > 0$

$$\lambda(k+1) = \alpha\lambda(k), \quad \lambda(1) = c. \tag{3.7}$$

As mentioned in the literature, Equation (3.3) or Equation (3.4) plays a role to avoid the search to be stuck in a local minima. Heuristic algorithm on the contrary provides only local minimum solution.

Simulated Annealing Algorithm

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random channels allocation. $k = 1$.

S2 Random select a reader, say i. $k = k + 1$.

S3 If reader's channel assignment has no collision to its neighbor readers, then goto S2.

S4 If reader's channel assignment has collision to its neighbors, random generate a new $\tau_i^* \in \{1, 2, \cdots, T\}$.

S5 If the number of collision pairs between the reader and its neighbors reduces, $\tau_i = \tau_i^*$.

S6 If the number of collision pairs between the reader and its neighbors increases by Δ, generate a random number u from a uniform distribution in $[0, 1]$. Then,

$$\tau_i = \begin{cases} \tau_i^* & \text{if } u \leq \exp(-\Delta/\lambda(k)), \\ \tau_i & \text{if } u > \exp(-\Delta/\lambda(k)). \end{cases} \quad (3.4)$$

S7 Repeat steps S2 to S6 for $k \leq MaxRun$.

Figure 6: Simulated annealing algorithm

3.1.3 Distributed Color Selection

Distribute Color Selection (DCS) resembles the heuristic algorithm mentioned before. Except that the new channel being selected for a reader if collision has encountered is random, see Step S4 in Figure 7.

3.2 Progressive approach for dynamic number of channels

Prior to search for a channel assignment, the number of channels (i.e. T) for allocation is usually not available. Owing not to guess the number for T, an algorithm that can automatically determine the number will be an advantage. One approach is to progressively increase the number of available channels after each cycle of search. Suppose the total number of available channels is initially set to a small integer, say 4. Then, apply either heuristic or simulated annealing algorithm to search for a solution for an allocation. If the solution obtained is not collision-free, increment the number of T by one and then re-run the algorithm. The steps are repeated until a collision-free solution is obtained. Follow this simple idea, five progressive type of algorithms can be applied. Four of them extended from the heuristic and SA-based algorithms. We call them progressive heuristic and progressive SA algorithms. The fifth algorithm is the Colorwave.

Distributed Color Selection Algorithm

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random channels allocation.

S2 Random select a reader, say i.

S3 If reader's channel assignment has no collision to its neighbor readers, then goto S2.

S4 If reader's channel assignment has collision to its neighbors, randomly select a new $\tau_i^* \in \{1, 2, \cdots, T\}$ other than the previous assignment.

S5 Repeat steps S2 to S4 until no more improvement can be made.

Figure 7: Distributed color selection.

3.2.1 Heuristic and SA-Based Algorithms

Figure 8 and 9 show the steps of the progressive heuristic and SA-based algorithms. Their working principles are not complicated. All the collided readers are assigned with the new channel $T+1$ if the solution obtained for the case when the number of channels is set to T is not collision-free. Then, the search is re-run with this new solution as the initial condition for the number of channels is set to $T+1$. The steps are repeated until a collision-free solution is obtained. If the solution obtained is again not collision-free, the collided readers are assigned with the channel $T+2$ and so on until a collision-free allocation is obtained.

3.2.2 Colorwave

Colorwave algorithm is an extension of the Distributed Color Selection (DCS) algorithm. For the DCS algorithm, the maximum number of available channels for assignment, i.e. T, is fixed. For Colorwave, this number is not fixed in advance. In Colorwave, each reader maintains a local T^i. Whenever a reader has collision, it selects randomly a new τ_i in $\{1, 2, \cdots, T^i\}$ as the channel for interrogation. The value of T^i can either be up or down by one, depended on the successful rate of assignment. If collision persists for a number of trial assignments, the value of T^i will be incremented by one. On the other hand, if collision-free assignment has succeeded for many times, the value of T^i will be decremented by one. The value of T will be maintained as a global variable indicating the maximum value of T^i in the network, i.e. $T = \max_i\{T^i\}$. A summary of the Colorwave algorithm is listed in Figure 10.

For illustration, Figure 11 shows the first three steps of a simple update example. The order of update is assumed to be from the Reader-4 to the Reader-3 and then to the Reader-7. The values inside the bracket correspond to the channel being assigned (τ_i) and the local maximum number of color (T^i). As the assignment of the Reader-4 has no collision, the reader simply set T^2 to 2. For Reader-3, it finds that there is no collision-free channel for selection as $T^3 = 3$. It then increases the value of T^3 by one (i.e. $T^3 = 4$) and sets τ_3 to 4. After that, it kicks this new assignment ($\tau_3 = 4$) to all its neighbors and sends the new T^3 to the central computer. For Reader-7, it finds its old assignment collides with its neighbor. It then

Progressive Heuristic Algorithm

S0 T is initialized to a small integer.

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random channels allocation.

S2 Random select a reader, say i.

S3 If reader's channel assignment has no collision to its neighbor readers, then goto S2.

S4 If reader's channel assignment has collision to its neighbors, select a new $\tau_i^* \in \{1, 2, \cdots, T\}$ such that the number of collision pairs between the reader and its neighbors is the minimum.

S5 Repeat steps S2 to S4 until no more improvement can be made.

S6 Check if the solution obtained is collision-free.

S7 Assign all the collided readers with a number $T + 1$.

S8 Repeat S2 to S7 with $T = T + 1$, until the solution is collision-free.

Figure 8: Progressive heuristic algorithm.

randomly selects amongst channels 1, 2 and 4 for the new assignment. Suppose it selects 1 for the new assignment (i.e. $\tau_7 = 1$) and this new assignment is of value smaller than its local maximum number of channel. Reader-7 then sets $T^7 = 1$ as the new local maximum color and kicks this new information to all its neighbors. We assume that the central computer broadcasts the readers the global maximum number of color (i.e. T) in the beginning of each cycle. This message also serves as a signal to synchronize all the readers.

4. Performance

To compare the performance of the algorithms, we base on an artificial RFID reader network, Figure 12. Within an area of $100m \times 100m$, 250 readers are deployed uniformly random. Two readers are neighborhood if their distance apart is less than or equal to 15m, i.e. $r \leq 15$. Each node in the figure corresponds to reader. The edges represent the neighborhood structure amongst the readers. The number of edges in the experimental network is 3830. In average, each reader has 15.32 neighbors.

4.1 Non-progressive Approach

For comparison, four non-progressive algorithms, including the heuristic-based algorithm, three SA-based algorithms and the DCS algorithm are running independently with the same initial condition on number of available channels T, where $T = 4, 5, 6, \cdots, 16$. For the heuristic-based algorithm, the total number of iterations is fixed to 2000. The total number of iterations (i.e.

Progressive Simulated Annealing

S0 T is initialized to a small integer.

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random channels allocation. $k = 1$.

S2 Random select a reader, say i. $k = k + 1$.

S3 If the reader has no collision to its neighbor, then goto S2.

S4 If the reader has collision to its neighbors, random generate a new $\tau_i^* \in \{1, 2, \cdots, T\}$.

S5 If the number of collision pairs between the reader and its neighbors reduces, $\tau_i = \tau_i^*$.

S6 If the number of collision pairs between the reader and its neighbors increases by Δ, generate a random number u from a uniform distribution in $[0, 1]$. Then,

$$\tau_i = \begin{cases} \tau_i^* & \text{if } u \leq \exp(-\Delta/\lambda(k)), \\ \tau_i & \text{if } u > \exp(-\Delta/\lambda(k)). \end{cases} \tag{3.8}$$

S7 Repeat steps S2 to S6 for $k \leq MaxRun$.

S8 Check if the solution obtained is collision-free.

S9 Assign all the collided readers with a number $T + 1$.

S10 Repeat S2 to S9 with $T = T + 1$, until the solution is collision-free.

Figure 9: Progressive simulated annealing algorithm.

Colorwave Algorithm

S1 Generate random numbers in $\{1, 2, \cdots, T\}$ for $\tau_1, \tau_2, \cdots, \tau_N$ as their initial random channels allocation.

S2 For $i = 1, \cdots, N$, initialize $S_i = 0$ as the successful rate of assignment, $K_i = 0$ as the number of trials since last update of T^i.

S3 Random select a reader, say i. $K_i = K_i + 1$.

S4 If reader's channel assignment has no collision to its neighbor readers, $S_i = S_i + 1$.

S4.1 If $S_i/K_i >$ DnSafe and $\tau_i < T^i$, set (i) $T^i = T^i - 1$ and (ii) $S_i = K_i = 0$.

S4.2 If $S_i/K_i >$ DnSafe and $\tau_i = T^i$, search for all collision-free channels for assignment. If the smallest numbered collision-free channel, τ_i^* is of number smaller than (i) T^i, $T^i = \tau_i^*$, (ii) $\tau_i = \tau_i^*$ and (iii) $S_i = K_i = 0$.

S5 If reader's channel assignment has collision to its neighbors, $S_i = S_i$.

S5.1 If $S_i/K_i <$ UpSafe, set (i) $T^i = T^i + 1$, (ii) $\tau_i = T^i + 1$ and (iii) $S_i = K_i = 0$.

S6 Update $T = \max\{T^i\}$.

S7 Repeat steps S2 to S6 until no more improvement can be made.

Figure 10: Colorwave algorithm.

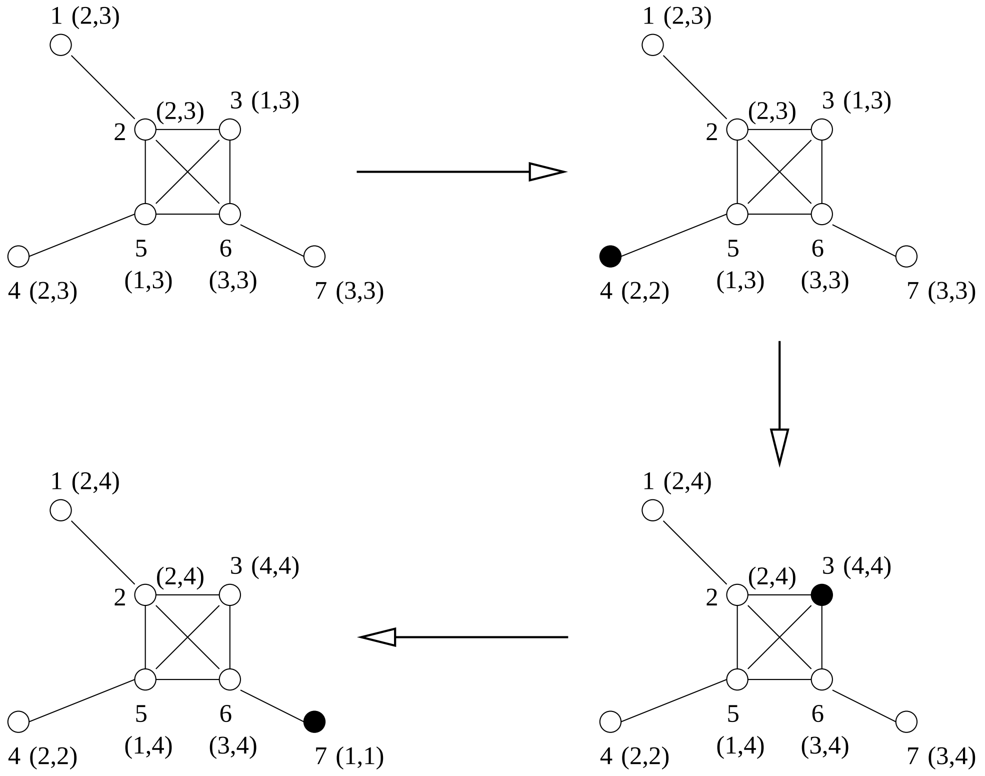


Figure 11: Illustrative example for the Colorwave algorithm.

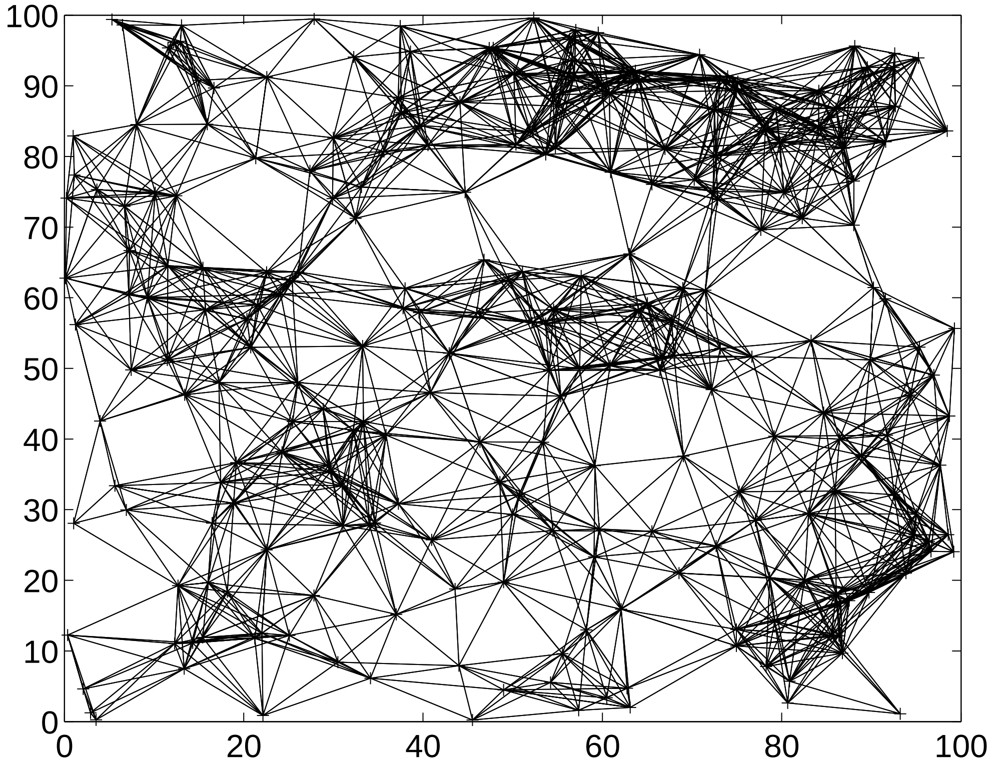


Figure 12: 250 readers in random locations within an area of 100m×100m. Readers at the end of an edge are neighbors. In average, each reader has 15.32 neighbors.

Table 1: Average number of collision pairs being counted for non-progressive algorithms.

Available channels	RAND	HEU	CT	GE	KP	DCS
4	497	255.4	229.6	230	228	498
5	384	170	141	143.2	143.6	403
6	309	112.8	91.8	93	92.8	333
7	262	81.6	59.2	61	60.2	272
8	229	52.2	37.2	37.2	38.4	224
9	192	35.4	23.2	23	22.6	216
10	183	21.4	11.8	12.4	12.2	195
11	157	12.2	6.6	6.8	6.8	175
12	146	7	3.4	3.4	3.6	158
13	144	3.6	1	1.2	1	149
14	121	1.8	0	0	0.2	122
15	127	0.4	0	0	0	115
16	115	0	0	0	0	97

RAND: Initial Random Allocation, HEU: Heuristic Algorithm
CT: Constant Temperature SA, GE: Geman-Geman Rule
KP: Kirkpatrick *et al* Rule, DCS: Distributed Color Selection.

$MaxRun$) for SA-based algorithms is 50000. These numbers are determined after a few trial runs. The constants a, b, c and α in the SA-based algorithms are 0.01, 1, 2 and 0.99.

4.1.1 Collision pairs

Again, two neighbor readers are in collision if their channels for interrogation are the same. Since the selection of a reader to update is random in each step, each experiment is repeated 5 times to avoid bias. The average number of collision pairs (defined in Equation (2.3)) being counted for different T are depicted in the following table.

The column with label RAND corresponds to the total number of collision pairs being counted right after the initial random allocation. The data shown in this column is for reference. One can see that all three simulated annealing algorithms perform similar to each other. Their solutions are slightly better than the heuristic algorithms. To have collision-free allocation, heuristic algorithm will need to have 16 channels in a cycle. While the SA algorithms employing constant temperature and the Geman-Geman rules require only 14 channels.

The DCS algorithm fails to generate a collision-free allocation, even for the number of available number channels is set to 16. As a matter of fact, the DCS algorithm has also been applied with 20 channels but the result is still negative. No collision-free allocation has been generated. It is believe that the failure is due to the fact that DCS does not take into the account of the neighborsassignment. Whenever a reader has encountered collision, it only selects randomly an alternative channel. In such case, even infeasible channels will be selected and collision can easily happen again. Therefore, the performance of DCS algorithm is so bad.

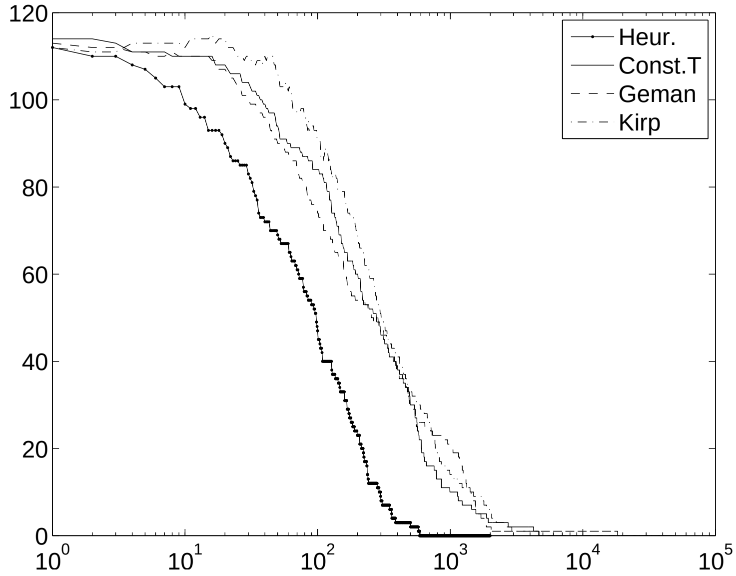


Figure 13: Number of reader collision pairs against the number of iterations. The total number of channels is 16.

4.1.2 Convergence

In terms of convergence speed, heuristic algorithm converges much faster than the simulated annealing algorithms. It only takes 600 iterations to complete the search. All three SA algorithms will need to take around 20000 iterations. Figure 13 shows a typical convergence plot of all four algorithms for the case when T is 16. Note that the horizontal axis is in logarithm scale.

4.1.3 Channels distribution

As mentioned before, each reader once interrogation has been finished will have to send the data to the control computer in the next channel. In this regards, a large number of readers assigned with the same channel will lead to a large volume of data transmission within a short time. Congestion and data lost might happen in return. To alleviate this effect, one method is to evenly allocate the channels to the readers.

Figure 14 shows the channel distributions generated by the algorithms in one experiment. The result for the DCS algorithm is not shown in the figure, as it fails to generate a collision-free solution. The distributions are obtained by the following steps. Let $\tau_1, \tau_2, \cdots, \tau_N$ be the channel assignment. The number of readers being assigned with the k^{th} channel is counted. That is, for $k = 1, 2, \cdots, T$,

$$f_k = \sum_{i=1}^{N} A(i,k), \quad A(i,k) = \begin{cases} 1 & \text{if } \tau_i = k \\ 0 & \text{otherwise.} \end{cases}$$

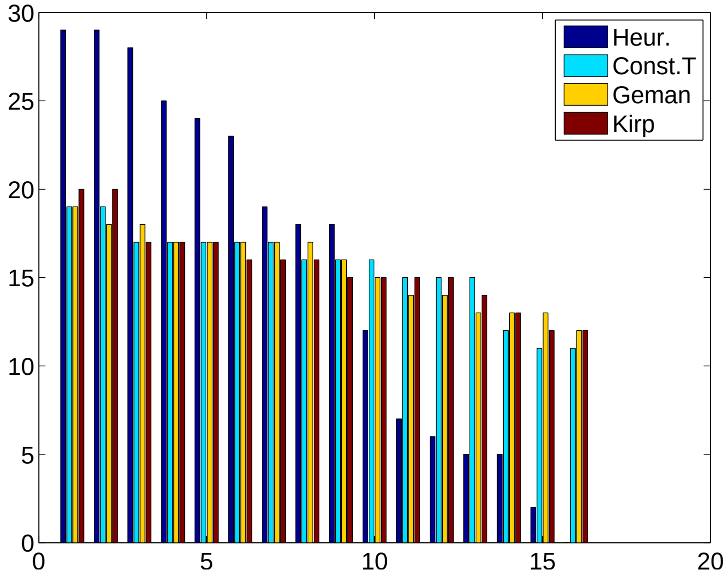


Figure 14: Number of readers assigned to different channels for non-progressive algorithms. The total number of channels is 16.

Then, the values of f_k (for $k = 1, 2, \cdots, T$) are sorted in descending order, i.e.

$$f_{\pi_1} \geq f_{\pi_2} \geq \cdots \geq f_{\pi_T}.$$

The values of f_{π_k} against k are plotted in the figure.

It is clear that the channel assignment obtained by the heuristic algorithm is not evenly distributed. It concentrates in a few channels. The maximum number of readers assigned to a channel is 29. For some channels, fewer than 5 readers are assigned. On the other hand, the distributions for the SA-based algorithms are rather even. The number ranges from 11 to 20. Their corresponding entropies, given by

$$\text{Entropy} = -\sum_{k=1}^{T} \frac{f_{\pi_k}}{N} \log\left(\frac{f_{\pi_k}}{N}\right) \tag{4.1}$$

are evaluated and depicted in Table 2. For reference, the entropy for 16 channels in which the readers are evenly assigned is 2.7726.

4.2 Progressive Approach

For the progressive approach, the simulations are carried out with the same reader network and the initial total number of available channels (i.e. T) is set to 4. Other conditions are the same as for the non-progressive simulations. The number of iterations for each round in the SA-based simulation is set to 10000. For the Colorwave algorithm, two experiments in which T equals to 4 and 20 have been conducted. In both simulations, the number of iterations is set to 5000 and their convergence behaviors are shown in Figure 15.

Table 2: Entropies of the channel distributions for non-progressive algorithms.

Algo.	Entropy	No. of channels
Heuristic	2.5249	16
SA CT	2.7605	16
SA GE	2.7633	16
SA KP	2.7622	16

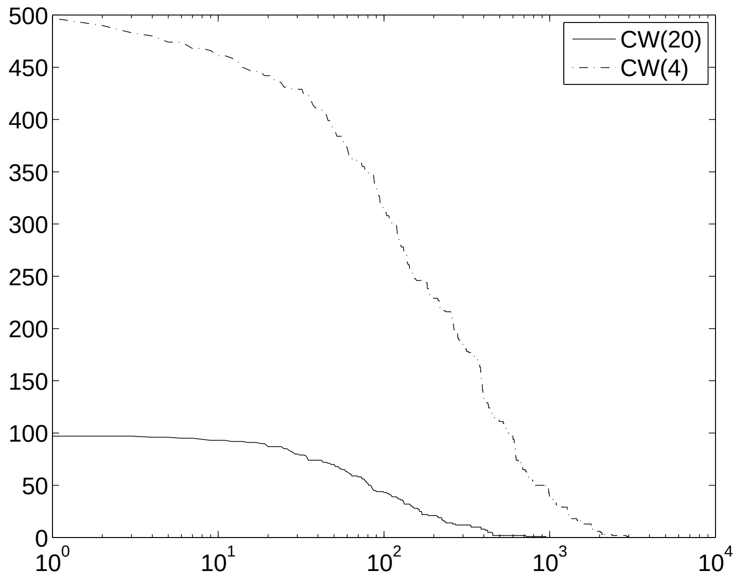


Figure 15: The convergence behavior of Colorwave with T equals 4 and 20 .

Table 3: Total number of channels required for a collision-free allocation using the progressive algorithms and Colorwave.

Algo.	Initial 4 channels
P. Heuristic	15-16
PSA CT	14-15
PSA GE	15-16
PSA KP	14-15
Colorwave	15-16

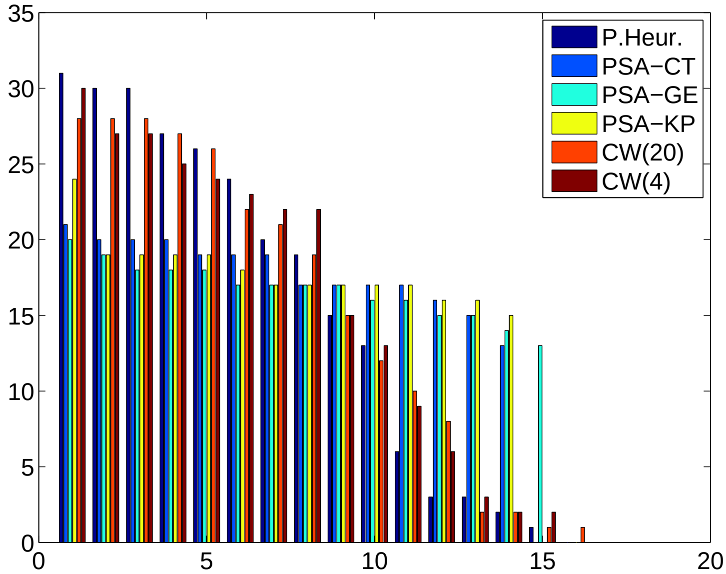


Figure 16: channels distributions generated by different progressive algorithms.

The total number of channels required for a collision-free allocation is depicted in Table 3. It is clear that the number of channels required for a collision-free allocation is similar for all algorithms, from 14 to 16. Two progressive simulated annealing algorithms perform slightly better. While the progressive heuristic, the PSA-GE and the Colorwave algorithms need more channels for collision-free allocation. The values of f_{π_k} against k are shown in Figure 16. Their entropies are depicted in Table 4. For reference, the entropies for 14 channels in which the readers are evenly assigned is 2.6391. While the entropies for 15 channels in which the readers are evenly assigned is 2.7081.

One can see that assignments obtained by progressive heuristic and the Colorwave algorithms are similar. Most readers are concentrated in a few channels $(2.4453 - 2.5040)$. While those obtained by progressive SA-based algorithms are more evenly distributed $(2.6325 - 2.7020)$.

Table 4: Entropies of the channel distributions for the progressive algorithms and the Colorwave (with initial 20 channels and initial 4 channels).

Algo.	Entropy	No. of channels
P. Heuristic	2.4453	15
PSA CT	2.6315	14
PSA GE	2.7020	15
PSA KP	2.6325	14
CW(20)	2.5040	16
CW(4)	2.4997	15

5. Fault Tolerance

In the above studies, we assume that all the readers are working perfectly. However, hardware deterioration could cause a reader malfunction. Here, we model this malfunction as a random channel flip. Once it occurs, it is likely to have new reader collisions and the collided readers are assumed to stop interrogation. Therefore, we could like to investigate the performance of the channel assignments that are generated by the heuristic algorithm, simulated annealing algorithms and the Colorwave algorithm. We measure their performance by counting the number of collided readers. The smaller the number of collided readers, the better the performance.

5.1 Expected number of collided readers

Suppose the k^{th} reader is fault and its original channel number of i. As $p \ll 1$, we can assume that none of its neighbors and the neighbors of its neighbors is fault, see Figure 17. Suppose Reader B is fault. We can assume that Reader A is functioning normally as the probability that two neighbor readers inside the circle centered at Reader B are fault is very small. By the same reason, we can assume that Reader C is functioning normally as Reader B and Reader C are inside the circle centered at Reader A. Once Reader B is fault, the probability that Reader C is fault is very small. Once the j^{th} is fault, it randomly flips to an unknown channel. If it flips to channel 1, it will collide the neighbors with channel 1. Then, the expected number of collided readers for this case will be $(q_1 M_k + 1)$, where M_k is the number of neighbors of the k^{th} reader and

$$q_i = \frac{\text{No. of Readers Assigned to Channel } i}{N}. \tag{5.1}$$

Similarly, the expected number of collided readers if it flips to channel j and $j \neq i$ is $(q_j M_k + 1)$.

As a result, the expected number of collided readers within the k^{th} reader and its M_k neighbors is given by

$$\begin{aligned} R_{Ck} &= \frac{1}{T} \sum_{j=1, j\neq i}^{T} (q_j M_k + 1) \\ &= \frac{1}{T} \left(T - 1 + (1 - q_i) M_k \right). \end{aligned} \tag{5.2}$$

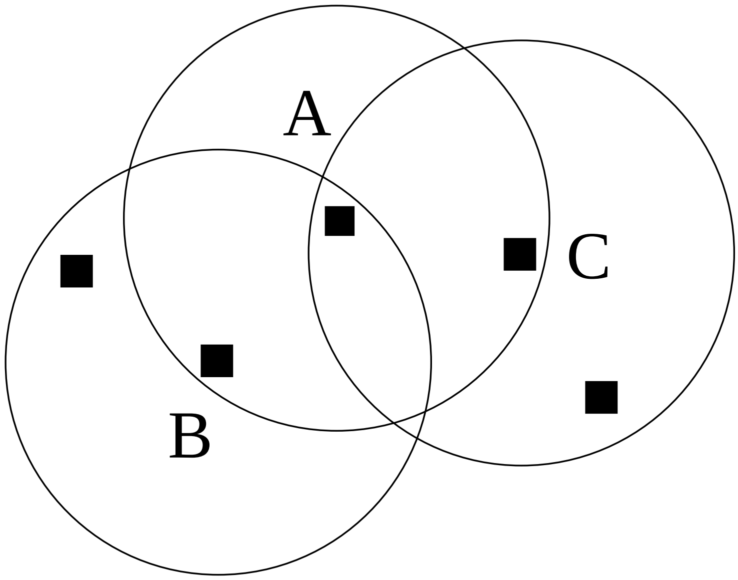


Figure 17: If Reader B is fault, the probability that Reader A and Reader C function normally is very high.

Taking expectation of $\mathcal{R}_{Ck}$ over all possible M_k, the expectation of R_{Ck} (denoted by $E[R_{Ck}|i]$) is given by

$$E[R_{Ck}|i] = \frac{1}{T}\left(T - 1 + (1 - q_i)M\right), \tag{5.3}$$

where M is the average number of reader neighbors in the reader network. Taking expectation of (5.3) over all possible i, we get that

$$\begin{aligned} E[R_{Ck}] &= \sum_{i=1}^{T} q_i E[R_{Ck}|i] \\ &= \frac{1}{T}\sum_{i=1}^{T} q_i \left(T - 1 + (1 - q_i)M\right) \\ &= 1 - \frac{1}{T} + \frac{M}{T} - \frac{M}{T}\sum_{i=1}^{T} q_i^2. \end{aligned} \tag{5.4}$$

Let $\mathcal{R}_C$ be the expected number of collided readers in the network given that the reader fault rate is p.

$$\begin{aligned} \mathcal{R}_C &= \sum_{k=1}^{N} pE[R_{Ck}] \\ &= pN\left[1 - \frac{1}{T} + \frac{M}{T} - \frac{M}{T}\sum_{i=1}^{T} q_i^2\right]. \end{aligned} \tag{5.5}$$

Now, we are interested in the maximum value of $\mathcal{R}_C$. As $\sum_{i=1}^{T} q_i = 1$, we get that $q_T = 1 - \sum_{i=1}^{T-1} q_i$. Hence, we can get that

$$
\begin{aligned}
F(q_1, q_2, \cdots, q_{T-1}) &= \sum_{i=1}^{T} q_i^2 \\
&= \left[1 - \sum_{i=1}^{T-1} q_i\right]^2 + \sum_{i=1}^{T-1} q_i^2. \qquad (5.6)
\end{aligned}
$$

Clearly, $F(q_1, q_2, \cdots, q_{T-1})$ is quadratic. It has one and only one equilibrium point.

Taking the first and second derivatives of $F(q_1, q_2, \cdots, q_{T-1})$ with respect to $q_1, q_2, \cdots, q_{T-1}$, we get that

$$
\begin{aligned}
\nabla F &= 2\left(\begin{bmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 2 \end{bmatrix}\begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{T-1} \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}\right). \qquad (5.7) \\
\nabla\nabla F &= 2\begin{bmatrix} 2 & 1 & \cdots & 1 \\ 1 & 2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 2 \end{bmatrix} \qquad (5.8)
\end{aligned}
$$

From (5.7), it is clear that $\nabla F = 0$ if $q_1 = q_2 = \cdots = q_{T-1} = 1/T$. Besides, we can show that $\nabla\nabla F$ is positive definite. Therefore,

$$F(q_1, q_2, \cdots, q_{T-1}) \geq \frac{1}{T},$$

i.e. $\sum_{i=1}^{T} q_i^2 \geq 1/T$. Thus by (5.5),

$$\mathcal{R}_C \leq pN\left(1 - \frac{1}{T}\right)\left(1 + \frac{M}{T}\right). \qquad (5.9)$$

Equality holds when $q_1 = q_2 = \cdots = q_T = 1/T$. On the other hand, one should note that $\sum_{i=1}^{T} q_i^2 \leq 1$ for all non-negative q_is and $\sum_{i=1}^{T} q_i = 1$. A lower bound for $\mathcal{R}_C$ in (5.5) is given by

$$\mathcal{R}_C \geq pN\left(1 - \frac{1}{T}\right) \qquad (5.10)$$

Combining (5.9) and (5.10), we get the bounds for the expected number of collided readers as follows :

$$pN\left(1 - \frac{1}{T}\right) \leq \mathcal{R}_C \leq pN\left(1 - \frac{1}{T}\right)\left(1 + \frac{M}{T}\right). \qquad (5.11)$$

5.1.1 Example 1: Hexagonal cell structure

Let us have an example to show the use of Equation (5.5). Suppose the readers are deployed in regular hexagonal structure. In such case, each reader will have six neighbors. Three channels will be enough collision avoidance. Therefore, for large N, $q_1 = q_2 = q_3 = 1/3$.

$$\mathcal{R}_C = pN \quad \left. 1 - \frac{1}{3} + \frac{6}{3} - \frac{6}{3} \sum_{i=1}^{3} \frac{1}{3^2} \right)$$

which equals to $2pN$. In other words, the utilization rate is $(1 - 2p)$. For a network consists of 250 readers and their fault rate is 0.05, the average number of collided readers is 25.

5.1.2 Example 2: Triangle channel distribution

Applying the heuristic and Colorwave algorithms, we have observed from the simulations that the channel distributions resemble triangle distribution. Let T be the number of available channels. Without loss of generality, we assume that

$$q_1 < q_2 < \cdots < q_T, \quad \text{and} \quad q_{i+1} = q_i + \Delta q.$$

For $\sum_{i=1}^{T} q_i = 1$, we can have the following equality.

$$Tq_1 + \frac{T(T-1)}{2}\Delta q = 1,$$

and hence

$$\Delta q = \frac{2(1 - Tq_1)}{T(T-1)}.$$

With reference to the simulation results, we can further let $q_1 = 1/250$. Thus,

$$\Delta q = \frac{2(250 - T)}{250T(T-1)}.$$

For $T = 16$, $\Delta q = 0.0078$ and

$$\mathbf{q} = (\begin{array}{llll} 0.0040, & 0.0118, & 0.0196, & 0.0274 \\ 0.0352, & 0.0430, & 0.0508, & 0.0586 \\ 0.0664, & 0.0742, & 0.0820, & 0.0898 \\ 0.0976, & 0.1054, & 0.1132, & 0.1210 \end{array}).$$

As a result, $\sum_{i=1}^{16} q_i^2 = 0.0832$. Again, from the simulation, we have found that the average number of neighbors is 15.32. For 0.05 fault rate, we can get that

$$\mathcal{R}_C = 0.05(250) \left(1 - \frac{1}{16} + \frac{15.32}{16} - \frac{15.32}{16} \times 0.0832 \right).$$

Therefore, the average number of collided reader is around 22.7.

5.1.3 Example 3: Uniform channel distribution

For uniform channel distribution, in which we have observed from the simulated annealing algorithms, we can get that

$$\mathcal{R}_C = 0.05(250)\left(1 - \frac{1}{16} + \frac{15.32}{16} - \frac{15.32}{16^2}\right).$$

Therefore, the average number of collided reader is around 22.94. Compared with Triangle distribution, one can see that their difference is insignificant.

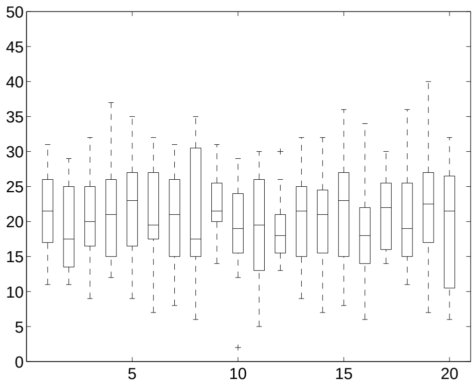
5.2 Simulations

Examples in the previous subsection have shown how to applied (5.5) to estimate the average number of collided readers (i.e. $\mathcal{R}_C$) for three different structure. Now, we are going to see confirm the equation (5.11) for the range of $\mathcal{R}_C$. The following simulations are conducted. 20 random reader networks are generated. Each network is generated in a similar manner as in Figure 12. Within an area of $100m \times 100m$, 250 readers are deployed uniformly random. Two readers are neighborhood if their distance apart is less than or equal to 15m. After a collision-free channel assignment has been found, each reader will randomly set to fault with 0.05 fault rate. Here, we assume that the total number of channels available for the heuristic algorithm and the simulated annealing algorithms is 16. For Colorwave, it does not have channel limit.

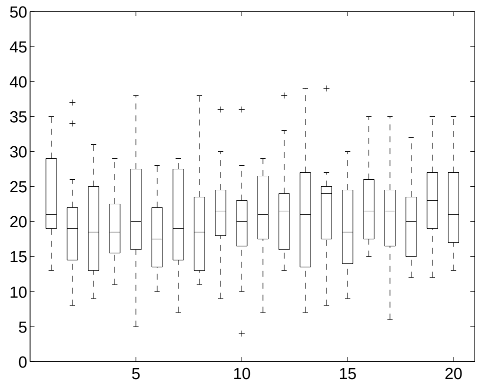
Figure 18 shows the simulation results of the non-progressive type algorithms. For all 20 networks, the number of collided readers ranges from 12 to 30. This range is roughly the same as the expected number of collided readers plus/minus 8. Figure 19 shows the simulations of Colorwave algorithm. As the channel assignment generated by Colorwave depends on the initial condition, we further repeated the simulations with different initial conditions. The diagrams on the left hand side show the number of channels required. Similar to the results of the non-progressive approach, the number of collided readers ranges from 10 to 30. This range is again roughly the same as the expected number of collided readers plus/minus 8.

6. Conclusion

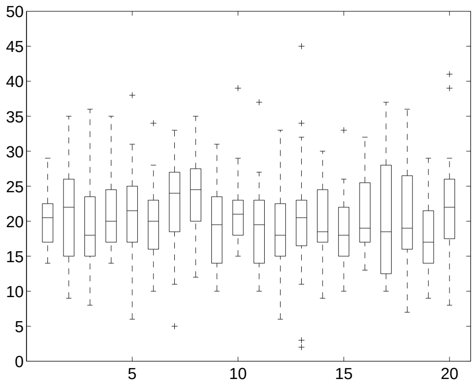
In this paper, a number of algorithms for solving reader collision problem have been elucidated. They include two heuristic-based algorithms, six SA-based algorithms, the DCS algorithm, the Colorwave algorithm and a hybrid heuristic-SA algorithm. In which, five of them are non-progressive type. The maximum number of channels for interrogation is predefined. Six of them are progressive type. The number of channels for interrogation is determined automatically by the algorithms. Their algorithms have been reviewed and their performance in the senses of their convergence, channel distribution and fault tolerance are presented. It is found that DCS fails to solve the channel assignment problem for random reader network. For the non-progressive approach, heuristic and SA-based algorithms can produce collision-free solutions. But, heuristic algorithm will need slightly more number than the SA-based algorithms and SA-based algorithms need much longer time to obtain the solutions. For the progressive approach, heuristic algorithm, SA-based algorithms and Colorwave are all able to determine the



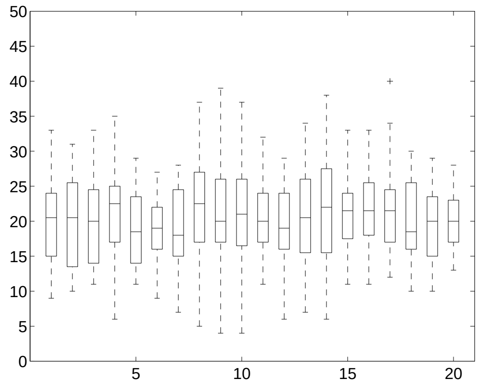
(a) Heuristic



(b) SA Constant Temp.



(c) SA Geman-Geman



(d) SA Kirkpatrick

Figure 18: The number of collided readers if fault rate is 0.05. (a) Heuristic algorithm. (b) SA algorithms employing constant temperature rule. (c) SA algorithms employing the Geman-Geman rule. (d) SA algorithms employing the Kirkpatrick rule. Here, the total number of channels is 16.

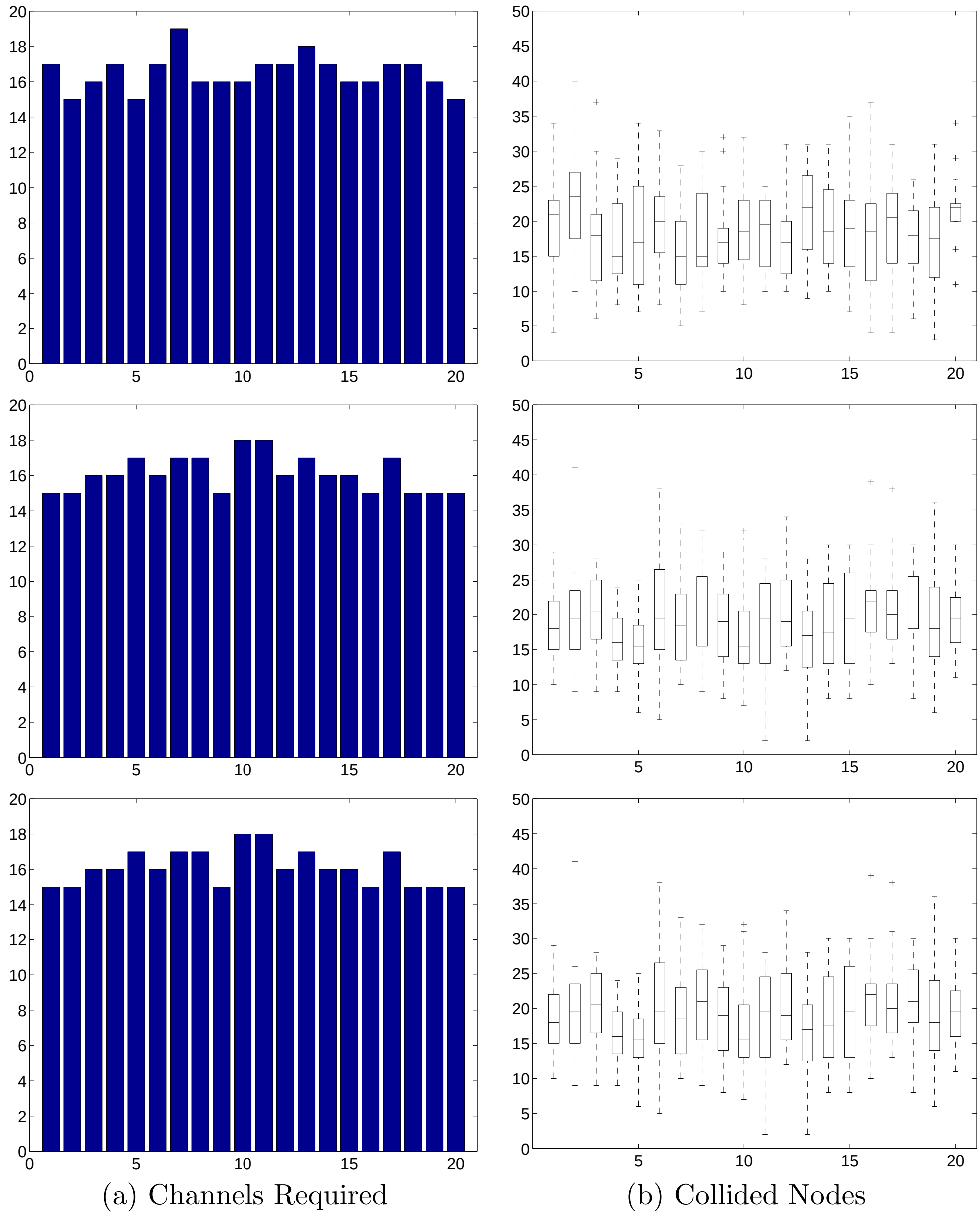


Figure 19: The number of collided readers if fault rate is 0.05 for Colorwave algorithm. (a) Number of collided readers. (b) The number of channels required. Three different trials give three slightly different results. Some results indicate that the number of channel required is larger than 16.

number automatically and simultaneously generate collision-free solution. However, SA-based algorithms require much longer time than the heuristic algorithm and Colorwave. Colorwave requires more channels. For channel distribution, SA-based algorithms can generate nearly even distributed channel assignments. Finally, analytical solution for the expected number of collided readers is derived for the condition that some readers in the network could flip randomly to other channel. Suppose this random flip can only happen after the channel assignment has finished. Together with the results on channel distributions, we found that the channel assignments obtained by SA-based algorithms are of least fault tolerance. All the above results indicate that heuristic algorithm could be a good choice for use in RFID readers channel assignment.

References

[1] C. Abraham, V. Ahuja, A.K. Ghosh and P. Pakanati, Inventory Management using Passive RFID Tags: A Survey, Department of Computer Science, the University of Texas at Dallas, Richardson, Texas, October, 2002.

[2] AutoID Center, 13.56 MHz ISM Band Class 1 Radio Frequency Identification Tag Interference Specification: Candidate recommendation, Version 1.0.0, Technical Report MIT-AUTOID-WH-002, MIT Auto ID Center, 2003.

[3] S. M. Birari, S. Iyer, Mitigating the reader collision problem in RFID networks with mobile readers, *IEEE Conference on Communication*, Vol. 1, page: 6, 16-18 Nov. 2005

[4] C. Boyd, Social Networking for Dogs, in MIT Technology Review, May 03, 2007. (http://www.technologyreview.com /Infotech/18628/)

[5] D.W. Engels, S.E. Sarma, The reader collision problem, in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, Vol. 3, pp.6, 2002.

[6] EPC Global, EPC Generation 1 Tag Data Standards Version 1.1 Rev.1.27, EPC Global Technical Report, May 2005.

[7] Etsi, ETSI EN 302 208-1 V1.1.1 September 2004. [Online] www.etsi.org.

[8] D. Krebs and M.J. Liard, Global Markets and Applications for Radio Frequency Identification, Venture Development Corporation White Paper, 2001.

[9] J. Ho, Daniel W. Engels, and Sanjay E. Sarma, HiQ: A Hierarchical Q-Learning Algorithm to Solve the Reader Collision Problem, *SAINTW'06*, 2006.

[10] Y.H. Huang, S.W. Hung and J. Sum, Simple algorithms for the assignment problem in structure reader networks, in *Proccedings of Internation Conference in Advanced Information Technologies*, Chaoyang University of Technology, April 25-26 2008, Taiwan.

[11] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, Optimization by simulated annealing, *Science*, vol. 220, no. 4598, pp. 671V680, 1983.

[12] J.K.Lee,S.K.Das and K.Kim, Analysis of RFID Anti-Collision Algorithms using Smart Antennas, *SenSys'04*, Baltimore, Maryland, USA, pp. 265-266, 2004.

[13] C.F. Lin and F.Y.S. Lin, 2007, A Simulated Annealing Algorithm for RFID Reader Networks, *Proceedings IEEE WCNC'07*, pp.1669 - 1672, March 2007.

[14] C.H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*, 2 Edition, Dover, 1998.

[15] T. S. Rappaport, *Wireless Communications- Principles and Practice*, 2nd Edition, Prentice Hall, 2002.

[16] S. Sarma, D. Brock, and D. Engels, Radio Frequency Identification and the Electronic Product Code, *IEEE Micro*, vol. 21, no.6, pp.50-54, 2001.

[17] D. H. Shih, P. L. Sun and D. C. Yen, Taxonomy and Survey of RFID Anti-Collision Protocols, *Computer communications*, Vol.29, No.11, pp.2150-2166. 2006.

[18] W. Stallings, *Wireless Communications & Networks*, Second Edition, 2005.

[19] W. Stallings, R. Want, An Introduction to RFID Technology, *IEEE Pervasive Computing*, vol.5, no.1, pp. 25-33, Jan.-March 2006

[20] J. Waldrop, D.W. Engels, S.E. Sarma, Colorwave: an anticollision algorithm for the reader collision problem, *IEEE Wireless Communications and Networking Conference*, pp.1206-1210, March, 2003.

[21] J. Waldrop, D.W. Engels, S.E. Sarma, Colorwave: A MAC for RFID reader networks, *IEEE Wireless Communications and Networking Conference*, pp. 1701-1704, 2003.

[22] R. Want. *RFID Explained: A Primer on Radio Frequency Identification technologies*, Morgan and Claypool, 2007

[23] R. Weinstein, J. Landt, The history of RFID, *IEEE Potentials*, Vol.24, no.4, pp. 8-11, Oct-Nov 2005

[24] T. Zhijun, and H. Yigang, Research of Multi-access and Anti-collision Protocols in RFID Systems, *Proc. IEEE Anti-counterfeiting, Security, Identification* pp.377-380, 16-18, April 2007.