

# Dynamic RFID Network Optimization Using a Self-adaptive Bacterial Foraging Algorithm

Hanning Chen\*, Yunlong Zhu, Kunyuan Hu, Tao Ku

Key Laboratory of Industrial Informatics, Shenyang Institute of Automation, Chinese Academy of Sciences,  
Shenyang, 110016, China  
{chenhanning; ylzhu; hukunyan; kutao}@sia.cn

## ABSTRACT

*Due to the low cost and simple deployment, radio frequency identification (RFID) systems are used in a variety of applications to uniquely identify physical objects. However, a number of critical design issues relating to dynamic determination of the optimal placement of RFID readers remain unresolved. In this paper, we address two important problems associated with reader location in RFID networks: (i) reader collision avoidance; (ii) optimal tag coverage. This paper, therefore, aims to use a successful swarm intelligence technique called self-adaptive bacterial foraging optimization (SABFO) to optimize both the reader-to-reader interference and tag coverage problems in RFID networks. The optimization results are also compared with the classical bacterial foraging optimization (BFO) algorithm. Numerical results for a RFID network case with ten readers have been presented to demonstrate the performance of the SABFO algorithm.*

**Keywords:** Dynamic RFID network optimization, reader collision, tag coverage, BFO, SABFO.

**2000 Mathematics Subject Classification:** 68T20.

## 1. Introduction

Radio frequency identification (RFID) technology is a type of automatic identification system. The purpose of an RFID system is to enable data to be transmitted by a portable device, called a tag, which is read by an RFID reader and processed according to the needs of a particular application. In recent years, an enormous amount of technical and commercial development of RFID has been demonstrated in many industrial applications, such as production, logistics, supply chain management and asset tracking (Wan, 1999; Park et al., 2006; Eric and Fred, 2008). In many applications, on the other hand, convenient, remote ad hoc deployment of wireless readers poses two challenging problems:

- **Reader collision (interference) avoidance:** One problem, reader collision (Engels and Sarma, 2002), occurs when co-located readers are simultaneously active. Specifically, reader collisions occur at tags situated in the vicinity of two or more readers that simultaneously interrogate tags. Such tags may be unable to correctly decode reader queries, leading to faulty or miss reads.
- **Optimal tag coverage:** Some applications, such as retail inventory tracking, need complete coverage only periodically. For such applications, using mobile readers to cover the area

---

\* Corresponding author. Address: Shenyang Institute of Automation, Chinese Academy of Sciences, Faculty Office III, Nanta Street 114#, Dongling District, Shenyang, 110016, China.

E-mail address: [chenhanning@sia.cn](mailto:chenhanning@sia.cn).

would be more cost-effective. However, before deploying the readers, it is necessary to answer many important questions, such as: (i) how many readers are needed for providing complete coverage, (ii) where should the readers be placed. This is the optimal tag coverage problem.

Thus, given an area to be covered completely within a period, determining the optimal placement and movement pattern of mobile readers and accurately detecting RFID tags in the presence of reader interference, is a difficult dynamic optimization problem.

In this work, we use a recently developed swarm intelligence search technique, namely the self-adaptive bacterial foraging optimization (SABFO) (Chen et al., 2008), to solve the problems of reader collision avoidance and optimal tag coverage in RFID systems. In stead of the simple description of chemotactic behavior in original bacterial foraging optimization (BFO) algorithm (Passino, 2002), SABFO also incorporates the adaptive search strategy, which allows each bacterium strikes a good balance between exploration and exploitation during algorithmic execution by tuning its run-length unit self-adaptively. Simulation results on the test scenario for a RFID network with ten readers are encouraging. That is, SABFO has superior search performance when compared to the original BFO.

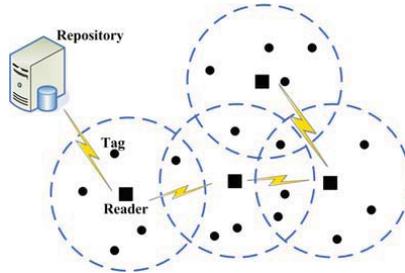


Fig. 1. An example RFID network deployment

The rest of the paper is organized as follows. In Section 2, RFID system models and the dynamic RFID network optimization problem definitions are presented. Section 3 first gives a review of the original BFO algorithm, and then proposes the novel SABFO. Section 4 describes the detailed design algorithm for dynamic RFID network optimization based on SABFO. Then the simulation results of a RFID network test scenario are discussed in Section 5. Finally, conclusions are drawn in Section 6.

## 2. Dynamic Optimization for RFID Network

The key components of an RFID system are the tags and readers. Both tags and readers have an antenna for radio communication with each other (Dong et al., 2008). The RFID tag, which is attached to the item to be tracked, stores the unique identification number of the item using a small integrated circuit. The RFID readers communicate with the tags by reading/writing the information stored on them. The reader has a limit on its interrogation range, within which the tags can be read. For example, RFID readers operating in the UHF band typically have an interrogation range of 3-5

m. As a result, the RFID reader network deployment is required to provide complete coverage for a given area with a large number of tags. The architecture of such an RFID system is illustrated in Fig. 1, where a central repository can gather data from readers through multi-hop wireless communication. In the figure, readers are represented by square nodes, tags are represented by round nodes, and the interrogation zone of a reader is represented by circles centered around readers. From a practical point of view, the geographical working area on which the RFID network is deployed is considered as a flat square surface.

In most practical RFID systems, the number of tags and their positions are highly dynamic and may change across very small time intervals (Anusha and Sridhar, 2005). Here we analyze such dynamic RFID system by using mobility models which define the pattern of tag movement. Note that in these mobility models, the number and the position of RFID tags change rapidly as tags enter and leave the system. We can use the following mobility model to capture the dynamics of a RFID network:

$$\begin{aligned}
 x_{step} &= \phi \cos(t) \\
 y_{step} &= \varphi \sin(t) \\
 X_i(t) &= X_i(t-1) + x_{step} \gamma(t) \\
 Y_i(t) &= Y_i(t-1) + y_{step} \tau(t)
 \end{aligned} \tag{1}$$

where  $(X_i, Y_i)$  represents the coordinate of the  $i^{th}$  tag in the working area,  $t$  is the time step,  $x_{step}$  and  $y_{step}$  are the step sizes in  $x$  and  $y$  direction respectively,  $\phi$  and  $\varphi$  are both a constant used for severity parameters,  $\gamma$  and  $\tau$  can be assigned 1, 0, or -1 with probability 1/3, respectively. Then in such dynamic RFID network, the system concerns two principal requirements and formulated as follows.

The first RFID system requirement is the optimal tag coverage, which is most important in RFID system (Guan et al., 2006). In this paper, if the radio signal received at a tag is higher than the threshold  $P_d = -10\text{dBm}$ , the communication between reader and tag can be established. Then the function is formulated as the sum of the difference between the desired power level  $P_d$  and the actual received power  $P_i'$  of each tag in time step  $t$ :

$$\text{Minimize } f_c(t) = \sum_{i=1}^{N_T} (P_i'(t) - P_d) \tag{2}$$

where  $N_T$  is the number of tags in the working area. This objective function is in order to locate the RFID readers close to the regions where the desired coverage level is higher, while the areas requiring lower coverage are taken into account by the proper radiate power increases of the readers.

Second, RFID reader collision (or interference) avoidance is considered in the RFID system. Reader interference mainly occurs in RFID network environment when several readers try to interrogate tags at the same time in the same area. This leads to an unacceptable level of misreads. Here we consider the sum of the received signals at each tag  $i$ , except the best served signal by reader  $k$ , as the interfering source. It can be defined below:

$$\text{Minimize } f_c(t) = \sum_{i=1}^N \left( \sum_{j=1}^M P_{ij}^r(t) - P_{ik}(t) \right) \quad (3)$$

where  $M$  is the number of readers,  $N$  is the number of tags in the working area, and  $P_{ij}^r$  is the power received from the  $j^{\text{th}}$  reader. The power level in this objective function is dBm. By changing the positions and radiated powers of readers, this objective function is in order to locate these readers far from each other to reduce interference levels.

In this paper, the overall optimal solution for RFID network optimization is represented by a linear combination of the two objective functions:

$$\text{Minimize } f_{all} = \kappa f_c + (1-\kappa)f_i, \quad \kappa \in [0,1] \quad (4)$$

This work set  $\kappa=0.5$ . However, it should be noted that the optimal weight value of the combined function can be varied according to different system requirements. In this work, each objective functions for RFID network optimization should be normalized because these objectives represent non-homogeneous quantities and are very different in values.

### 3. Self-adaptive Bacterial Foraging Optimization

#### 3.1 The Original BFO Algorithm

The original bacterial foraging optimization (BFO) algorithm is one of the state-of-the-art evolutionary algorithms (Badamchizadeh et al., 2010; Zhao et al., 2010), which consists of three principal mechanisms, namely chemotaxis, reproduction, and elimination-dispersal (Passino, 2002). We briefly describe each of these processes as follows:

##### a) Chemotaxis

In the original BFO, a unit walk with random direction represents a “tumble” and a unit walk with the same direction in the last step indicates a “run”. Suppose  $\theta^i(j, k, l)$  represents the bacterium at  $j^{\text{th}}$  chemotactic,  $k^{\text{th}}$  reproductive, and  $l^{\text{th}}$  elimination-dispersal step.  $C(i)$  is the chemotactic step size during each run or tumble (run-length unit). Then in each computational chemotactic step, the movement of the  $i^{\text{th}}$  bacterium can be represented as

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (5)$$

where  $\Delta(i)$  is the direction vector of the  $j^{\text{th}}$  chemotactic step. When the bacterial movement is *run*,  $\Delta(i)$  is the same with the last chemotactic step; otherwise,  $\Delta(i)$  is a random vector whose elements lie in  $[-1, 1]$ .

With the activity of run or tumble taken at each step of the chemotaxis process, a step fitness, denoted as  $J(i, j, k, l)$ , will be evaluated.

**b) Reproduction**

The health status of each bacterium is calculated as the sum of the step fitness during its life, i.e.  $\sum_{j=1}^{N_c} J(i, j, k, l)$ , where  $N_c$  is the maximum step in a chemotaxis process. All bacteria are sorted in reverse order according to health status. In the reproduction step, only the first half of population survives and a surviving bacterium splits into two identical ones, which are then placed in the same locations. Thus, the population of bacteria keeps constant.

**c) Elimination and Dispersal**

The chemotaxis provides a basis for local search, and the reproduction process speeds up the convergence which has been simulated by the classical BFO. While to a large extent, only chemotaxis and reproduction are not enough for global optima searching. Since bacteria may get stuck around the initial positions or local optima, it is possible for the diversity of BFO to change either gradually or suddenly to eliminate the accidents of being trapped into the local optima. In BFO, the dispersion event happens after a certain number of reproduction processes. Then some bacteria are chosen, according to a preset probability  $P_{ed}$ , to be killed and moved to another position within the environment. In what follows we briefly outline the original BFO algorithm step by step:

**[Step 1]** Initialize parameters  $n, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C(i)$  ( $i=1,2,\dots,S$ ),  $\theta^i$ . Where,

$n$ : Dimension of the search space,

$S$ : The number of bacterium,

$N_c$ : chemotactic steps,

$N_s$ : swim steps,

$N_{re}$ : reproductive steps,

$N_{ed}$ : elimination and dispersal steps,

$P_{ed}$ : probability of elimination,

$C(i)$ : the run-length unit. (i.e. the chemotactic step size during each run or tumble).

**[Step 2]** Elimination-dispersal loop:  $l = l+1$ .

**[Step 3]** Reproduction loop:  $k = k+1$ .

**[Step 4]** Chemotaxis loop:  $j = j+1$ .

[substep a] For  $i = 1=1, 2, \dots, S$ , take a chemotactic step for bacteria  $i$  as follows.

[substep b] Compute fitness function,  $J(i, j, k, l)$ .

[substep c] Let  $J_{last} = J(i, j, k, l)$  to save this value since we may find better value via a run.

[substep d] Tumble: Generate a random vector  $\Delta(i) \in R^n$  with each element  $\Delta_m(i), m = 1, 2, \dots, S$ , a random number on  $[-1, 1]$ .

[substep e] Move: Let

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{6}$$

This results in a step of size  $C(i)$  in the direction of the tumble for bacteria  $i$ .

[substep f] Compute  $J(i, j+1, k, l)$  with  $\theta^i(j+1, k, l)$ .

[substep g] Swim:

- (i) Let  $m = 0$  (counter for swim length).
- (ii) While  $m < N_s$  (if have not climbed down too long)
  - Let  $m = m+1$ .
  - If  $J(i, j+1, k, l) < J_{\text{last}}$ , let  $J_{\text{last}} = J(i, j+1, k, l)$ . then another step of size  $C(i)$  in this same direction will be taken as equation (1) and use the new generated  $\theta'(j+1, k, l)$  to compute the new  $J(i, j+1, k, l)$ .
  - Else let  $m = N_s$ .

[substep h] Go to next bacterium ( $i+1$ ): if  $i \neq S$  go to (b) to process the next bacteria.

[Step 5] If  $j < N_c$ , go to step 3. In this case, continue chemotaxis since the life of the bacteria is not over.

[Step 6] Reproduction:

[substep a] For the given  $k$  and  $l$ , and for each  $i = 1, 2, \dots, S$ , let

$$J_{\text{health}}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l) \quad (7)$$

be the health of the bacteria. Sort bacterium in order of ascending values ( $J_{\text{health}}$ ).

[substep b] The  $S_r$  bacteria with the highest  $J_{\text{health}}$  values die and the other  $S_r$  bacteria with the best values split and the copies that are made are placed at the same location as their parent.

[Step 7] If  $k < N_{re}$  go to step 2. In this case the number of specified reproduction steps is not reached and start the next generation in the chemotactic loop.

[Step 8] Elimination–dispersal: For  $i = 1, 2, \dots, S$ , with probability  $p_{ed}$ , eliminate and disperse each bacteria, which results in keeping the number of bacteria in the population constant. To do this, if a bacterium is eliminated, simply disperse one to a random location on the optimization domain. If  $l < N_{ed}$ , then go to step 2; otherwise end.

**Table 1.** Pseudocode for the dynamic self-adaptive strategy.

---

```

1: FOR (each bacterium  $i$ ) IN PARALLEL
2:   IF (Criterion-1) then
3:      $C^i(t+1) = C^i(t) / \alpha$ ; //exploitation
4:      $\mathcal{E}^i(t+1) = \mathcal{E}^i / \beta$ ;
5:   ELSE IF (Criterion-2) then
6:      $C^i(t+1) = C_{\text{initial}}$ ; //exploration
7:      $\mathcal{E}^i(t+1) = \mathcal{E}_{\text{initial}}$ ;
8:   ELSE
9:      $C^i(t+1) = C^i(t)$ ;
10:     $\mathcal{E}^i(t+1) = \mathcal{E}^i(t)$ ;
11:  END IF
12: END FOR IN PARALLEL

```

---

In the context of original BFO model, the bacteria with large run-length unit have exploring ability (global investigation of the search place) while the bacteria with relatively small run-length unit have exploiting skill (the fine search around a local optimum) (Chen et al., 2008). However, it is difficult to decide which values of static run-length unit is best suited for a given problem. Hence, we introduce the preprogrammed change of this parameter during the evolution to balance exploration and exploitation, which results in the significant improvement in performance of the original algorithm.

### 3.2 The SABFO Algorithm

In the SABFO algorithm, we introduce an "individual run-length unit" to the  $i$ th bacterium of the colony and each bacterium can only modify the search behavior of itself by using the current status of its own. In this way, not only the position (solution vector) but also the run-length unit of each bacterium undergoes evolution, respectively. This individual-level self-adaptation may provide us with more accurate information about the search because it is the separate individuals that engaged in searching the solution space not just the whole colony. In the foraging process of SABFO model, each bacterium displays alternatively two distinct search states:

- (1) *Exploration* state, during which the bacterium employs a large run-length unit to explore the previously unscanned regions in the search space as fast as possible.
- (2) *Exploitation* state, during which the bacterium uses a small run-length unit to exploit the promising regions slowly in its immediate vicinity.

Each bacterium in the colony has to permanently maintain an appropriate balance between *Exploration* and *Exploitation* states by varying its own run-length unit adaptively. In SABFO, the adaptation of the individual run-length unit is done by taking into account two decision indicators: a fitness improvement (finding a promising domain) and no improvement registered lately (current domain is food exhausted). The criteria that determine the adjustment of individual run-length unit and the entrance into one of the states (i.e. *Exploitation* and *Exploration*) are the following:

- *Criterion-1*: if the bacterium discovers a new, promising domain, the run-length unit of this bacterium is adapted to another smaller one. Here "discovers a new promising domain" means this bacterium registers a fitness improvement beyond a certain precision from the last generation to the current. Following *Criterion-1*, the bacterium's behavior will self-adapt into *Exploitation* state.
- *Criterion-2*: if the bacterium's current fitness is unchanged for a number  $K_v$  (user-defined) of consecutive generations, then augment this bacterium's run-length unit and this bacterium enters *Exploration* state. This situation means that the bacterium searches on an un-promising domain or the domain where this bacterium focuses its search has nothing new to offer.

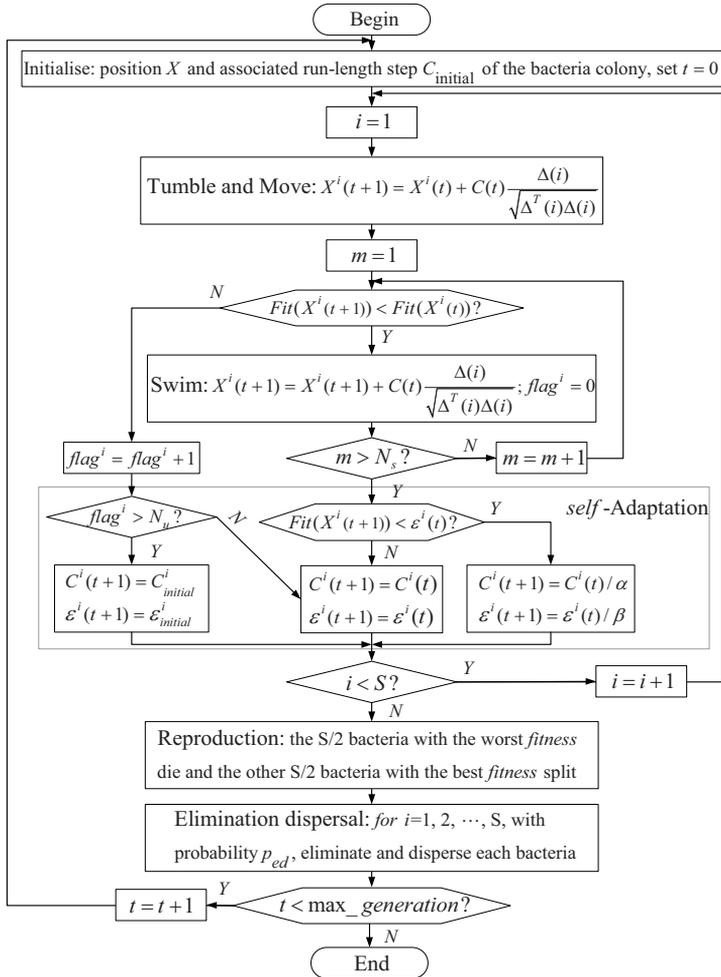


Fig. 2. Flowchart of the SABFO algorithm.

This self-adaptive strategy is given in pseudocode in Table 1. Where  $t$  is the current generation number,  $C^i(t)$  is the current run-length unit of the  $i$ th bacterium,  $\varepsilon^i(t)$  is the required precision in the current generation of the  $i$ th bacterium,  $\alpha$  and  $\beta$  are user-defined constants,  $C_{initial}$  and  $\varepsilon_{initial}$  are the initialized run-length unit and precision goal respectively.

The flowchart of the SABFO algorithm can be illustrated by Fig. 2, where  $S$  is the colony size,  $t$  is the chemotactic generation counter from 1 to max-generation,  $i$  is the bacterium's ID counter from 1 to  $S$ ,  $X^i$  is the  $i$ th bacterium's position of the bacteria colony,  $N_s$  is the maximum number of steps for a single activity of swim,  $flag^i$  is the number of generations the  $i$ th bacterium has not improved its own fitness.

#### 4. Dynamic RFID Network optimization base on SABFO

Then the overall operating process of dynamic RFID network optimization based on SABFO can be described as follows:

##### (1) Initialization Phase

###### a. Reader Specification

This gives the details of the mobile RFID readers that include the adjustable radiated power range; the according interrogation range - the distance up to which a tag can be read by the reader; the interference range - the distance within which if two readers transmit simultaneously their signals would interfere; and the number of the mobile reader to be used.

###### b. Topology Specification

This gives the details of the working area to be covered by RFID network according to the application scenario. It includes the shape and dimension of the region; the number of the RFID tags to be used; the tag distribution (i.e., the tag position) in the working area; and the tag power threshold – the minimum tag received power level under which the communication between reader and tag can be established.

###### c. Population Generation

Individuals forming the SABFO population should be randomly generated. Each bacterium is characterized by real number representation and has a dimension equal to  $3M$  ( $M$  is the number of used RFID readers),  $2M$  dimensionalities for the coordinates of reader positions (i.e., a possible network layout), and  $1M$  dimensionalities for radiated powers of each reader (i.e., a possible network parameter setting). For example, a real-number bacterium [12.55, 28.33, 16.78, 11.21, 25.11, 13.56, 0.50, 0.62, 1.73] is a possible planning solution of a RFID network containing 3 readers. The 1 to 6 bit means that the three readers are located in the two-dimension working area at (12.55, 28.33), (16.78, 11.21), and (25.11, 13.56) respectively. The 7 to 9 bits mean that the radiated power allocated to each reader is 0.50, 0.62, and 1.73(watt) respectively.

##### (2) Optimization Phase

At the end of the initialization phase, all the information needed for the optimization phase is obtained for generating the optimal RFID network topologies. The basic building blocks of this phase are:

###### a. Fitness Evaluation

For each bacterium, evaluate its fitness on a suitable optimizing objective function. Various optimizing objective functions are possible. That is, each function of the optimizing objective function set (Eqn. 2-4) for RFID network planning can be applied to meet different system requirements. For example, when deploys a RFID network in a supermarket, the most important thing is to find a solution that obtains the maximum tag reading rate. Then the objective function should be the maximum tag coverage objective function, namely Eqn. 2.

###### b. Population Evolution

Compare the evaluated fitness values and perform the chemotaxis, self-adaptation, reproduction, and elimination dispersal for each bacterium to update its position.

###### c. Termination condition

The computation is repeated until the maximum number of iteration is met or desired system requirements are obtained.

**Table 2.** The setting for an ideal RFID network example

Reader Specification		Topology Specification	
Reader	10	Dimension	30m×30m
Radiated	0.1 - 2watt	Tag number	100
Interrogation	3 - 4m	Dynamic tag	$\phi = \varphi = 0.1$
Interference	3.5 – 4.5m	tag power	-10dBm

**Table 3.** Parameters of the SABFO algorithm

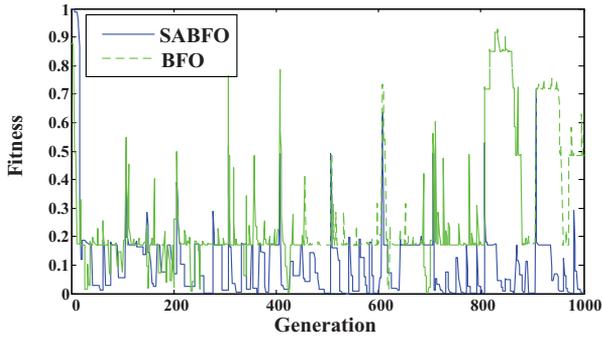
Func. Para.	S	$C_{initial}$	$\varepsilon_{initial}$	$K_u$	$\alpha$	$\beta$
$f_c$	50	1.0	100	20	10	10
$f_i$	50	1.0	100	20	10	10
$f_{all}$	50	1.0	100	20	10	10

### 5. Simulation Results

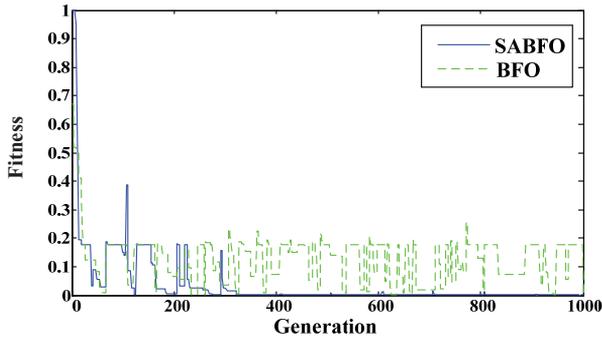
The readers used here are mobile and the tags are passive. we consider an ideal example shown in Table 2. that is, the proposed algorithm is evaluated against an ideal square working area: a 30m×30m working space with 100 tags that distributed dynamically. Ten RFID readers, whose radiated power is adjustable in the range from 0.1 to 2 watt, are considered to serve this area. Here the interrogation range according to the reader radiated power is computed as in literature (Dobkin, 2004). The real number solution representation is used in all the algorithms for solving the scheduling problem. When ten RFID readers are employed to serve the whole working area, this problem can be considered as a continuous optimization problem with 30 dimensions. Then each solution vector is characterized by 30 genes, namely 10+10 genes for (x, y) coordinates of reader positions and 10 genes for radiated powers of each reader.

The performance of SABFO on the dynamic RFID network optimization has been compared with the original BFO. The initialized population size and the maximum generation for all involved algorithms is 50 and 1000 respectively. The parameters settings for SABFO are summarized in Table 3. For the original BFO algorithm, we take  $C=0.1$  and  $P_{e0}=0.25$ , which is the standard set of these two parameter values as recommended in (Chen et al., 2010). We set  $S=100$ ,  $N_c=100$ ,  $N_s=4$ ,  $N_{re}=5$  and  $N_{ed}=2$ , then BFO performs totally 1000 chemotactic steps in each run, which make a fair comparison in regard of the parameter values. The comparison between SABFO and BFO on each objective function (Eqn. 2-4) is given in Fig. 3, where the best fitness in a single run are plotted to demonstrate the typical optimization behaviors. As it is shown in Fig. 3, the SABFO

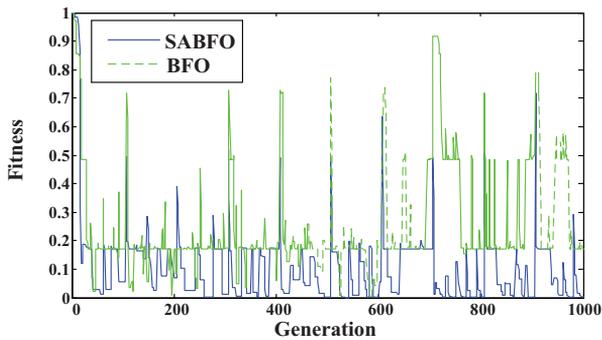
is able to track the time-variant global optima of each objective functions much more smoothly and effectively.



(a)  $f_c$

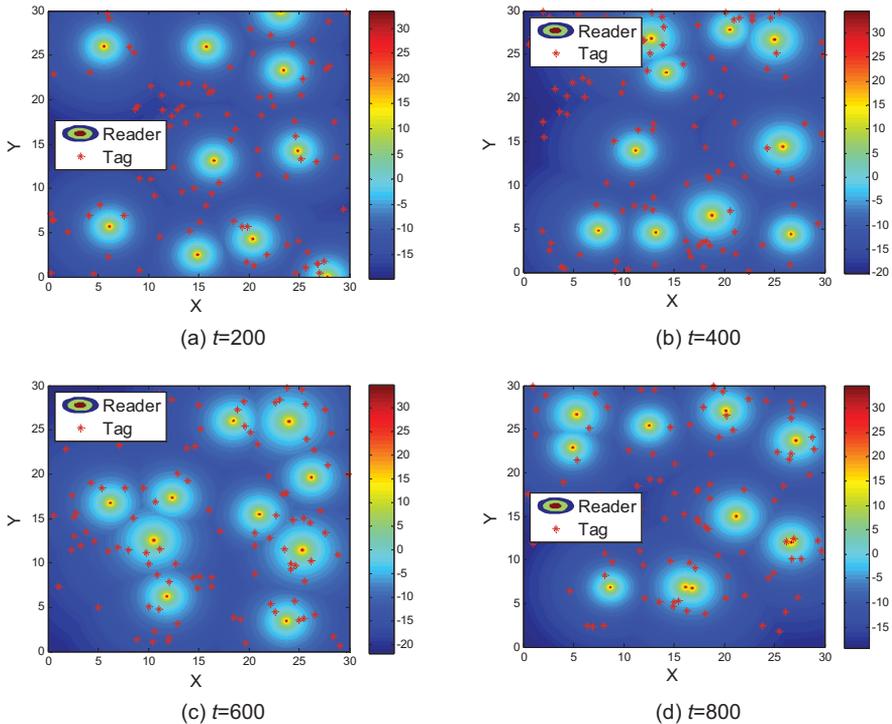


(b)  $f_i$



(c)  $f_{all}$

Fig. 3. Best fitness in the dynamic tag environment.



**Fig. 4.** Dynamic RFID network optimization results in different time steps.

To clearly illustrate the simulation results, Fig. 4 shows several RFID network optimization solutions obtained in different time steps by SABFO for the combined objective function. In each time step, the tag distribution and the corresponding locations of ten RFID readers optimized by SABFO is shown in Fig. 4(a)-(d), which also report the received power levels according to the path loss. As it is shown in the figure, the algorithm increase the powers and concentrate the position of readers in the regions of the working area where higher tag coverage is required; at the same time, the algorithm tries to provide sufficient limited distances between RFID readers.

## 6. Conclusions

In this paper, we develop a dynamic RFID network optimization model based on the bacterial foraging search methods. This proposed model regards the dynamic RFID network optimization problem as a multi-objective optimization problem, where the optimal tag coverage and reader collision avoidance objectives are particularly considered as the primary requirements of the RFID system.

In the case study, two bacterial foraging optimization methods, namely the original bacterial

foraging optimization (BFO) and self-adaptive bacterial foraging optimization (SABFO), are both employed to search for the optimal placements of ten mobile RFID readers in an ideal test scenario. The simulation results show that the SABFO obtains superior solutions than the original BFO method.

#### Acknowledgement

This work is supported by the National 863 plans projects of China under Grant 2008AA04A105.

#### References

- Anusha, S., Sridhar, I., 2005, *RFIDcover - A coverage planning tool for RFID networks with mobile readers*, Lecture Notes in Computer Science, Embedded and Ubiquitous Computing, 3823, 1047-1057.
- Badamchizadeh, M.A., Nikdel, A., Kouzehgar, M., 2010, *Comparison of genetic algorithm and particle swarm optimization for data fusion method based on Kalman filter*, International Journal of Artificial Intelligence, 5 (10), 67-78.
- Chen, H.N., Zhu, Y.L., Hu, K.Y., 2008, *Self-adaptation in bacterial foraging optimization algorithm*, Proceedings of the Intelligent System and Knowledge Engineering, Xiamen, China, 1026-1031.
- Chen, H.N., Zhu, Y.L., Hu, K.Y., 2010, *Multi-colony bacteria foraging optimization with cell-to-cell communication for RFID network planning*, Applied Soft Computing, 10 (2), 539-547.
- Dobkin, D. M., 2008, *The RF in RFID: passive UHF RFID in practice*, Elsevier, New York.
- Dong, Q.f., Shukla, A., Shrivastava, V., Agrawal, D., Banerjee, S., Kar, K., 2008, *Load balancing in large-scale RFID systems*, Computer Networks, 52 (9), 1782-1796.
- Engels, D.W., Sarma S.E., 2002, *The reader collision problem*, Proceedings of the 2002 IEEE International Conference on Systems, Man and Cybernetics, MIT, USA, 3, 370-376.
- Eric, N., Fred R., 2008, *RFID: Technology, applications, and impact on business operations*, International Journal of Production Economics, 112 (2), 507-509.
- Guan, Q., Liu, Y., Yang, Y. P., Yu, W. S., 2006, *Genetic approach for network planning in the RFID systems*, Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications, Beijing, China, 2, 567-572.
- Passino, K.M., 2002, *Biomimicry of bacterial foraging for distributed optimization and control*, IEEE Control System Magazine, 22(3), 52-67.
- Park, D., Choi, Y.B., and Nam, K.C., 2006, *RFID-based RTLS for improvement of operation system in container terminals*, Proceedings of the Asia-Pacific Conference on Communications, Busan, Korea, 1-5.
- Wan, D., 1999, *Magic medicine cabinet: A situated portal for consumer healthcare*, Proceedings of the International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Berlin/Heidelberg, Germany, 352-355.
- Zhao, Z.Y., Xie, W. F., Hong, H., 2010, *Hybrid optimization method of evolutionary parallel gradient search*, International Journal of Artificial Intelligence, 5 (10), 1-16