

# Using Parallel Particle Swarm Optimization For RFID Reader-to-reader Anti-collision

Bin Cao\*, Shan Yang, Jianwei Zhao  
Hebei University of Technology  
Tianjin 300401, China  
caobin@scse.hebut.edu.cn;  
yangshan1191@163.com

Po Yang\*, Atif Waraich  
Department of Computer Science  
Liverpool John Moores University  
Liverpool, L3 5UA, UK.  
poyangcn@gmail.com

**Abstract**—With the wide application of radio frequency identification (RFID) technology, the possibility of the collision among readers may increase. When the number of RFID readers is large, the dimension of the RFID reader collision problem will be huge. To solve the high-dimensional RFID reader-to-reader collision problem effectively, we improve the parallel cooperative co-evolution particle swarm optimization (PCCPSO) algorithm by adopting the hybrid adaptive strategy of the inertia weight. In addition, we make parallelism implementation of the improved algorithm. Then, we use the improved algorithm to solve the RFID reader-to-reader anti-collision problem. In the experiments, we compare the improved distributed parallel particle swarm optimization (IDPPSO) algorithm with the PCCPSO algorithm, and make Wilcoxon test on the results. The experimental results demonstrate IDPPSO algorithm has better performance.

**Keywords**—RFID reader-to-reader anti-collision, hybrid adaptive inertia weight, particle swarm optimization

## I. INTRODUCTION

RFID technique has been applied in many fields, such as vehicle networks [1] and [2], rail management [3] and [4], cargo tracking [5] [6], and student attendance monitoring [7] [11] [16], etc. However, as the application of the RFID system increases, it may lead to a large number of readers in the same deployment region. When the signal from a tag to a target reader encounters interference from the signal of another working reader, a reader-to-reader collision problem may occur. In the case of dense deployment of readers, the reader-to-reader collision problems will become serious. These affect the identification rate and the reliability of RFID systems [8] [20] [22].

A number of existing RFID reader anti-collision algorithms are based on resource scheduling strategy. The Distributed Color Selection (DCS) [9] protocol is a well-known strategy for the RFID reader collision problem. In this protocol, the reader which collides with others will randomly select a new time slot for communication. However, this protocol takes a long time to achieve good results. The work of [10] presented the Neighbor Friendly Reader Anti-collision (NFRA) protocol to manage time slots for readers; it uses “Listen Before Talk” strategy to reduce collisions among neighbors. NFRA is simple to implement, but it only considers the single channel situation and it needs extra hardware. Li et al. [21] proposed an RFID reader-to-reader anti-

collision model, in which the resources to be allocated are readers’ physical positions and transmitting power, as well as the time slots and frequency channels. They also proposed an artificial immune network algorithm with hybrid encoding for resource allocation (AINetHE-RA) to solve it. However, the AINetHE-RA algorithm performs poorly when handling high-dimensional (relatively large numbers of readers) RFID reader anti-collision problems.

A cooperative co-evolving particle swarm optimization (CCPSO) algorithm [12] was used to address the high-dimensional non-separable problems. CCPSO integrated strategies of random grouping and adaptive weighting. Experiments have shown that CCPSO performs well only in a small number of evaluations. Furthermore, in the work of [13], by introducing dynamic grouping and a novel position-updating rule, a new cooperative coevolving particle swarm optimization (CCPSO2) algorithm was presented. CCPSO2 algorithm performs well at solving the optimization problem when the dimensions are up to 2000. In the work of [14], a Spark-based parallel cooperative co-evolution PSO algorithm (PCCPSO) was proposed by drawing the advantage of CCPSO2. Experimental results show that the PCCPSO algorithm had good optimization performance in dealing with high-dimensional problems. Therefore, this motivates us to make improvements based on PCCPSO algorithm, and use it to solve the RFID reader anti-collision problem with relatively high dimensionality.

Based on the PCCPSO algorithm, we adopt the hybrid adaptive strategy of the inertia weight. In addition, to improve the computational efficiency of the algorithm, we use the MPI parallel programming interface to conduct the parallelism of the algorithm. Then, we use the improved distributed parallel PSO algorithm (IDPPSO) for solving the RFID reader-to-reader anti-collision optimization problem.

The remainder of this paper is organized as follows. Section II describes the RFID reader collision problem and the RFID reader-to-reader anti-collision model. Our proposed algorithm, IDPPSO, is presented in detail in Section III. In Section IV, we describe the experimental results. Finally, we make a conclusion of this paper in Section V.

\* Corresponding authors: Bin Cao, Po Yang.

## II. THE RFID READER-TO-READER ANTI-COLLISION MODEL

### A. The RFID Reader Collision Problem

The RFID reader collision problem includes reader-to-tag and reader-to-reader collisions. When two readers simultaneously send signals to the same tag in the overlapped interrogation region of the two readers, a reader-to-tag collision occurs. An example is presented in Fig. 1, where  $r$  is the interrogation radius of reader  $R_1$ , and  $x$  is the interference radius of reader  $R_2$ . The interrogation range of reader  $R_1$  is occluded by both the interrogation range and the interference range of reader  $R_2$ . Tag  $T_2$  is within the interrogation ranges of readers  $R_1$  and  $R_2$ . When readers  $R_1$  and  $R_2$  transmit signal to tag  $T_2$  at the same time, a reader-to-tag collision occurs, resulting in two readers read/write failures. Additionally, when the target reader is located within the interference range of a working reader, the signal of the working reader performs as the interference signal to the target reader a reader-to-reader collision occurs. As shown in Fig. 1,  $T_1$  is within the interrogation range of  $R_1$  and the interference range of  $R_2$ . When  $R_1$  transmits a signal to  $T_1$ , if  $R_2$  is also in an active state, its signal acts as an interference to  $R_1$ . Moreover, if the power of the interference signal makes the interrogation radius of  $R_1$  smaller than the distance between  $R_1$  and  $T_1$ , the read/write ( $R_1$  from/to  $T_1$ ) will fail.

### B. The RFID Reader-to-reader Anti-collision Model

In the work of [21], the readers' physical positions, transmitting power, time slots, and frequency channels are considered in a reader-to-reader anti-collision model. This model formulates the reader-to-reader anti-collision problem as an optimization problem of resource allocation to readers. The decision vector of the optimization problem contains the physical position of each reader, as well as frequency channel encoding and power encoding of each reader at each time slot. Therefore, the dimensionality of the candidate solution is:

$$\text{Dimensions} = N_{\text{reader}} \times 2 + N_{\text{reader}} \times N_{\text{slot}} \times 2, \quad (1)$$

where  $N_{\text{reader}}$  and  $N_{\text{slot}}$  are the numbers of readers and time slots, respectively. In this model, the communication between reader  $i$  and tag  $k$  can be established at time slot  $t$  only if the signal-to-interference-plus-noise ratio ( $\text{SINR}_i(t, x_{i,k})$ ) of reader  $i$  is greater than a desired minimum value  $\text{SINR}_{\text{min}}$ . Therefore, the interrogation radius of reader  $i$  at time slot  $t$  is defined as  $r_i^t$ , which can be calculated as follows:

$$r_i^t = \arg \max_{x_{i,k}} \text{SINR}_i(t, x_{i,k}) \geq \text{SINR}_{\text{min}}, \quad (2)$$

where  $x_{i,k}$  is the distance between reader  $i$  and tag  $k$ , the calculation formulation of  $\text{SINR}_i(t, x_{i,k})$  is detailed in reference [21].

Throughput is an important index for evaluating the performance of the RFID system. Therefore, based on the RFID reader-to-reader model proposed in [21], we consider the throughput of the RFID system in one interrogation round as the evaluating index of the model. One interrogation round includes  $N_{\text{slot}}$  time slots. In this paper, the throughput is defined as the number of effectively identified tags in one interrogation round

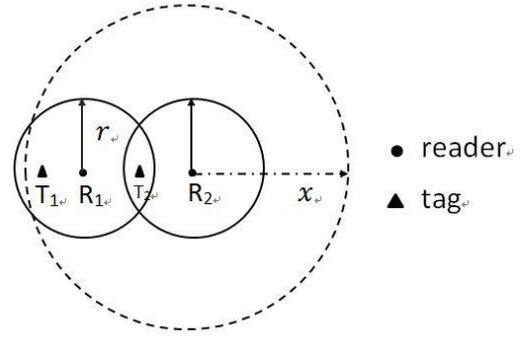


Fig. 1. Illustration of reader collision problem.

(suppose a certain number of tags randomly distributed in the specified deployment region). The calculation formula of the throughput is as follows:

$$\text{Throughput} = \sum_{k=1}^{N_{\text{tag}}} \delta_k, \quad (3)$$

where  $N_{\text{tag}}$  is the number of tags,  $\delta_k$  indicates whether tag  $k$  is effectively identified in the interrogation round,  $\delta_k = 1$  indicates tag  $k$  is effectively identified, and it is calculated as follows[15]:

$$\delta_k = \begin{cases} 1 & \exists t \in SS, \exists i \in SR, s.t. x_{i,k} \leq r_i^t, \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where  $SR = \{1, 2, \dots, N_{\text{reader}}\}$ ,  $SS = \{1, 2, \dots, N_{\text{slot}}\}$ .

## III. IMPROVED DISTRIBUTED PARALLEL PSO ALGORITHM BASED ON MPI

### A. The Improved PSO Algorithm

GPSO [19] algorithm has the advantage of fast convergence speed in solving unimodal problems, while the LPSO [17] algorithm can effectively avoid being trapped in local optima in multimodal problems. PCCPSO algorithm combined the GPSO algorithm and the LPSO algorithm, taking advantage of merits of both algorithms. Each particle contains the evolution information of both GPSO algorithm and LPSO algorithm. The two algorithms optimize independently, but in the optimization process, each algorithm will absorb advanced information of the other one. Before the position updating, GPSO algorithm updates its global optimal to the better one of the two algorithms' global optimal positions; in the process of LPSO algorithm position updating, LPSO algorithm updates its position to the global optimal of GPSO algorithm based on a specified probability. This paper improves the PCCPSO algorithm, which will be illuminated in detail in the following.

In the GPSO algorithm, particle position updating is formulated as follows:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_{h,i}(t) - p_i(t)) + c_2 r_2 (p_g(t) - p_i(t)), \quad (5)$$

$$p_i(t+1) = p_i(t) + v_i(t+1). \quad (6)$$

where  $t$  is the current iteration number;  $\omega$  is the inertia weight;  $c_1$  and  $c_2$  are learning factors;  $p_{h,i}$  is the best position of the  $i^{th}$  particle found so far,  $p_g$  is the best position found by the entire particle swarm so far,  $p_i$  is the current position of the  $i^{th}$  particle;  $v_i$  indicates the current velocity of the  $i^{th}$  particle;  $r_1$  and  $r_2$  are random numbers within  $[0,1]$ .

The inertia weight in PCCPSO algorithm is set to a constant. An adaptive inertia weight control strategy is proposed in the work of [18]. In this strategy, the inertia weight changes adaptively according to the optimization state of the algorithm. The inertia weight is calculated as follows:

$$\omega_t(f) = \frac{1}{1+1.5e^{-2.6f}}, f = \frac{d_g - d_{min}}{d_{max} - d_{min}}, \quad (7)$$

$$d_i = \frac{1}{n-1} \sum_{j=1, j \neq i}^n \sqrt{\sum_{k=1}^D (x_j^k - x_i^k)^2} \quad (8)$$

where  $D$  is the dimension of the decision vector;  $n$  is the population size;  $x_i^k$  is the  $k^{th}$  dimensional value of the decision vector;  $d_i$  is the mean distance between particle  $i$  and the other particles;  $d_{max}$  and  $d_{min}$  are the maximum and minimum distances, respectively;  $d_g$  is the mean distance of the global best particle;  $f$  is the evolutionary factor that tracks the optimization state of the algorithm.

In the work of [19], an inertia weight adaptive strategy of linearly decreasing with the iterative generations is proposed. In this paper, we combine the two kinds of inertia weight adaptive strategies, the hybrid adaptive strategy of the inertia weight in this paper can be defined as follows:

$$\omega_t = \begin{cases} (\omega^f - \omega^i) \frac{t}{T} + \omega^i, & \frac{t}{T} \leq s \\ \frac{1}{1+1.5e^{-2.6f}}, & otherwise \end{cases}. \quad (9)$$

here  $T$  is the total number of iterations;  $\omega^f$ ,  $\omega^i$  and  $s$  are constants. In the early stage of the evolution, the inertia weight decreases linearly according to generation. This strategy can make the population search in a wider range of the search space, enhancing the population's global search ability. In addition, when using the inertia weight linear decreasing strategy, it does not need to judge the optimization state of the algorithm in each iteration process, which takes a relatively short time and improves the efficiency of the algorithm. Moreover, when judging the optimization state of the algorithm according to equations (7) and (8), there may be a case of misjudgment. In the early stage of evolution, if inertia weight is changed according to the optimization state of the algorithm, it may cause a large change of the inertia weight because of the misjudgment. That may damage the coherence of the particle search. In the later stage of the evolution, the inertia weight adaptively changes according to the optimization state of the algorithm. When the algorithm is in the exploitation or convergence state, the value of the evolutionary factor  $f$  is relatively small, so the inertia weight will also be relatively small, and the local search capability of the population is enhanced. When the algorithm is

in the exploration state or jumping-out state (jump out of the local optimum), the value of the evolutionary factor  $f$  is relatively large, so the inertia weight will also be relatively large, which can enable the population to search wider in the search region and enhance the global search ability. This strategy can improve the diversity of population distribution and prevent the population from falling into local optimum. To sum up, by combining the two inertia weight adaptive strategies, the hybrid adaptive strategy can absorb the advantages of the two strategies and enhance the effectiveness of the algorithm.

#### B. Parallelism Implementation based on MPI

In dense deployment of RFID readers application scenarios, the dimension of the problem to be solved may be very high. When attempting to solve this kind of complex and high-dimensional optimization problem with the conventional serial method, the computation time may be very long. Therefore, to improve the computational efficiency of the proposed IDPPSO algorithm, we use the message passing interface (MPI) for parallel programming. In this paper, the parallelism has a coarse granularity: each process represents a particle and all of the processes execute in parallel. In the procedure of algorithm optimization, each process communicates with others through MPI to complete the information exchange among particles.

### IV. EXPERIMENTS

#### A. Experimental Setup

In the simulations, there are 1000 tags placed randomly in a  $100 \times 100 m^2$  rectangular region. The deployment area of readers is the same region. The parameters of the RFID reader-to-reader anti-collision model are as follows: the number of time slots is 4, the number of communication channels is 10, and the maximum power of reader's transmitting signal is 1.0 W; the other parameters of the model are the same as in [21]. In the experiment, we compare the effectiveness of the proposed IDPPSO algorithm and the PCCPSO algorithm when solving the RFID reader-to-reader anti-collision problem. In the proposed IDPPSO algorithm,  $\omega^f$ ,  $\omega^i$  and  $s$  are set to 0.4, 0.9, and 0.5, respectively. The parameters of the PCCPSO algorithm is in reference [14].

In the evaluations, we use the Tianhe-2 supercomputer as the experimental platform to test the proposed IDPPSO algorithm. There are twenty-four cores in each node, and we used forty cores of two nodes. We also use coarse-grained parallelism in the proposed algorithm, with each process representing a particle (i.e. the number of particles is forty). Each test instance is executed twenty times, and the averaged fitness value is taken to be the result of the instance. For fair comparison, the FEs (number of fitness evaluations) settings in the proposed IDPPSO algorithm and the PCCPSO algorithm are the same:  $Dimensions \times 10^3$ .

#### B. Comparison of the Algorithms Effectiveness

Here, we evaluate the effectiveness of the two algorithms in dealing with the RFID reader-to-reader anti-collision problem. We have considered five cases of RFID reader quantities, which are 20, 40, 60, 80, and 100, respectively. Fig. 2 illustrates the test results obtained by IDPPSO algorithm and PCCPSO algorithm. As shown in Fig.2, it can be seen that the proposed IDPPSO

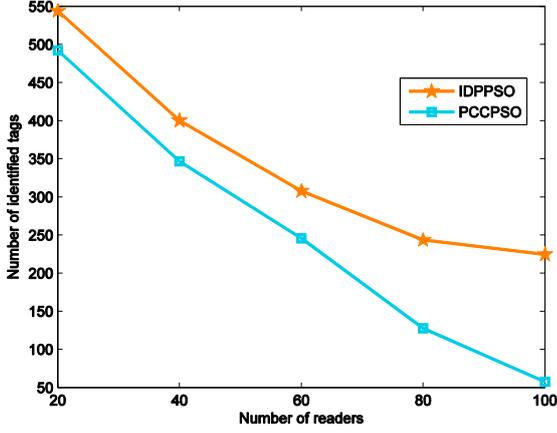
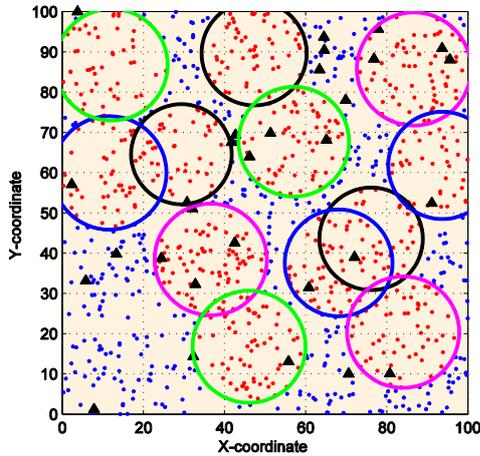


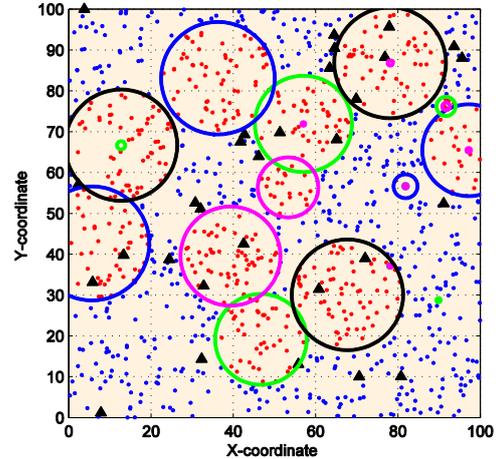
Fig. 2. Test results obtained by IDPPSO algorithm and PCCPSO algorithm.

algorithm outperforms the PCCPSO algorithm at all of the tested dimensionalities. In addition, it can be seen from Fig.2, as the dimension increases, the gap between the two algorithms is also increasing. That is to say, the proposed IDPPSO algorithm performs well particularly in dealing with high-dimensional optimization problems.

Fig. 3 depicts the deployment effect obtained by IDPPSO algorithm and PCCPSO algorithm when the number of readers is 20. In Fig. 3, the small black triangle represents the predefined candidate locations of readers; the small blue square represents the tag which is not effectively identified, and the red small square represents the tag which is effectively identified. As shown in Fig. 3, the four colors of the circle represent the four time slots (the representative colors of slots 1, 2, 3 and 4 are black, blue, green, and magenta, respectively). It can be seen from Fig. 3, the number of tags which are effectively identified in IDPPSO algorithm is larger than PCCPSO algorithm. That is



(a) The proposed IDPPSO algorithm



(b) The PCCPSO algorithm

Fig. 3. 20 RFID readers' test results obtained by the proposed IDPPSO algorithm and the PCCPSO algorithm.

TABLE I. WILCOXON TEST RESULT OF IDPPSO AND PCCPSO.

Algorithm	$R+$	$R-$	$p$ -value	$\alpha=0.05$
IDPPSO	\	\	\	\
PCCPSO	15.0	0.0	0.030971	yes

to say, the throughput obtained by the proposed IDPPSO algorithm is better than PCCPSO algorithm.

In addition, we perform the Wilcoxon test on the results in Fig. 2, and the statistical result is presented in Table I. Therefore, we can examine the significance of the difference between the results of the two algorithms. As shown in Table I, when IDPPSO algorithm is compared to PCCPSO algorithm, the  $R+$  value is greater than the  $R-$  value, and the  $P$ -value is less than 0.05. Therefore, we can conclude that IDPPSO algorithm is significantly better than PCCPSO algorithm at solving the RFID reader-to-reader anti-collision optimization problem in the tested dimensionalities.

### C. Comparison of the Computation Time

The proposed IDPPSO algorithm uses the MPI parallel programming interface to improve the computational efficiency. We tested the computing time of the parallel version and the serial version of the proposed algorithm. In the evaluations, we used the Tianhe-2 supercomputer as the experimental platform of the parallel and serial programs. In the parallel programs, we used forty cores. As shown in Table II, we compared the computing time between the parallel version and the serial version of the proposed algorithm. We tested their computing time when solving with the 600, 1000 and 2000 dimensional RFID reader-to-reader anti-collision problems. The speedups of the parallel version compared to the serial version are about 46.2%, 48.9%, and 82.6% with respect to the ideal speedup (i.e.,

TABLE II. THE COMPUTING TIME OF THE PARALLEL VERSION AND THE SERIAL VERSION OF IDPPSO.

Dimension	Parallel Version	Serial Version
600	2.28 E+01 s	4.21 E+02 s
1000	5.48 E+01 s	1.07 E+03 s
2000	1.92 E+02 s	6.35 E+03 s

40) at 600, 1000, and 2000 dimensionalities, respectively. Therefore, the parallel version of the proposed algorithm reduces a large amount of computing time and improves the efficiency of the algorithm.

## V. CONCLUSION

We apply the improved distributed parallel PSO algorithm to the RFID reader-to-reader anti-collision optimization problem. In the improved algorithm, we adopt the hybrid adaptive strategy of the inertia weight. To solve the high-dimensional RFID reader-to-reader anti-collision problem effectively and efficiently, we use the MPI parallel programming interface. We compared the improved algorithm with PCCPSO algorithm, and the findings demonstrated that the proposed IDPPSO algorithm performs significantly better than the PCCPSO algorithm. In addition, the experimental results show that the parallel strategy reduces a lot of computing time and improves the efficiency of the algorithm. The future work includes a refinement of the RFID reader anti-collision model and the improvement of the algorithm.

## ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61303001, in part by the Opening Project of Guangdong High Performance Computing Society under Grant No. 2017060101, in part by the Foundation of Key Laboratory of Machine Intelligence and Advanced Computing of the Ministry of Education under Grant No. MSC-201602A, in part by the Special Program for Applied Research on Super Computation of the NSFC Guangdong Joint Fund (the second phase) under Grant No. U1501501.

## REFERENCES

- [1] X. L. Cheng, J. Lu, and W. Cheng, "A survey on RFID applications in vehicle networks," Proc. 2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI), pp.146–151, (2015).
- [2] Y. X. Liu and L. Jia, "Research on vehicle actuated coordinated control method based on RFID electronic tag," Proc. 2015 8th International Symposium on Computational Intelligence and Design (ISCID), pp. 327–330, (2015).
- [3] W. He, Y.J. He, and M. M. Tentzeris, "Survey of RFID applications in railway industry," Proc. 2014 First International Conference on Automation, Control, Energy and Systems (ACES), pp. 1-6, (2014) .

- [4] J. H. Qiu, B. Sun, and Q. D. You, "Study on RFID antenna for railway vehicle identification," Proc. 2006 6th International Conference on ITS Telecommunications, pp. 237–240, (2006) .
- [5] E. A. Kadir, S. M. Shamsuddin, S. K. Abdul, and S. L. Rosa, "RFID middleware for fast clearance in container terminal management system", Proc. 2015 3th International Conference on Information and Communication Technology (ICoICT), pp. 478–481, (2015).
- [6] E. A. Kadir, S. L. Rosa, and H. Gunawan, "Application of RFID technology and e-seal in container terminal process," Proc. 2016 4th International Conference on Information and Communication Technology (ICoICT), pp. 1–6, (2016) .
- [7] R. B. Kuriakose and H. J. Vermaak, "Developing a Java based RFID application to automate student attendance monitoring," Proc. Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), pp.48-53, (2015).
- [8] L. C. Zhang, R. Ferrero, F. Gandino, and M. Rebaudengo, "Evaluation of single and additive interference models for RFID collisions," Mathematical and Computing Modelling, vol.58, pp.1236–1248, (2013).
- [9] J. Waldrop, D. W. Engles, and S. E. Sarma, "Colorwave: A MAC for RFID reader networks," in Proc. IEEE Conf. Wireless Commun. Netw., pp. 1701–1704, (2003) .
- [10] J. B. Eom, S. B. Yim, and T. J. Lee, "An efficient reader anti-collision algorithm in dense RFID networks with mobile RFID readers," IEEE Trans. Ind. Electron., vol. 56, no. 7, pp. 2326–2336, (2009).
- [11] P. Yang, "Improving Passive RFID Localisation Precision with Moving Direction Estimation based Feature Improvement", *IET Wireless Sensor Systems Journal*, Vol. 4, Issue, 1, Jan, 2014.
- [12] X. D. Li and X. Yao, "Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms," Proc. IEEE Congress on Evolutionary Computation, pp.1546-1553, (2009).
- [13] X. D. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," IEEE Transactions on Evolutionary Computation, vol.16, n0.2, pp.210-224, (2012).
- [14] B. Cao, et al. "Spark-based parallel cooperative co-evolution particle swarm optimization algorithm," Proc. IEEE International Conference on Web Services, pp. 570-577, (2016).
- [15] Z. Li , C. He, J. Li, and X. Huang, "RFID reader anti-collision algorithm using adaptive hierarchical artificial immune system," Expert Systems with Applications, vol. 41, no. 5, pp. 126-2133, (2014) .
- [16] P. Yang, W. Wu, "Efficient Particle Filter Localisation Algorithm in Dense Passive RFID Tags Environment", *IEEE Transactions on Industry Electronics* , Vol 61, Issue 10, pp. 5641 - 5651, Oct, 2014.
- [17] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," Congress on Evolutionary Computation, pp. 1671-1676, (2002).
- [18] Z. Zhan, J. Zhang, Y. Li, and H. S. Chung, "Adaptive particle swarm optimization," IEEE Trans. Syst. Man Cybernetics, Part B Cybern., vol. 39, no. 6, pp. 1362–1381, (2009).
- [19] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," IEEE World Congr. Comput. Intell, pp. 69-73, (1998).
- [20] P. Yang, W. Wu, M. Moniri and C. C. Chibelushi, "Efficient Object Localisation Using Sparsely Distributed Passive RFID Tags," *IEEE Transactions on Industry Electronics* , Vol 60, Issue 12, pp. 5914-5924, Dec, 2013.
- [21] Z. H. Li, J.M. Li, and C.G. He, "Artificial immune network-based anti-collision algorithm for dense RFID readers," Expert Systems with Applications, vol.41, no.10, pp.4798-4810, (2014) .
- [22] P. Yang, W. Wu, M. Moniri and C. C. Chibelushi, "SLAM Algorithm for 2D Object Trajectory Tracking based on RFID Passive Tags", *IEEE International Conference on RFID 2008*, Las Vegas, Nevada, USA on April 16-17, 2008.