RFID Network Scheduling Using a Discrete Multi-swarm Optimizer

Liu Wei, Niu Ben, Chen Hanning

1 Jilin Normal University, Siping, 136000, China, lwzxm1020@126.com
2 Laboratory of Information Service and Intelligent Control, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, 110016, China, chenhanning@sia.cn
3 College of Management, Shenzhen University, Shenzhen, 518060, China, Drniuben@gmail.com

Abstract

The operation of RFID systems often involves a situation in which multiple readers physically located near one another may interfere with one another’s operation. Such reader collision must be minimized to avoid the faulty or miss reads. Specifically, scheduling the colliding RFID readers to reduce the total system transaction time or response time is the challenging problem for large-scale RFID network deployment. This paper, therefore, aims to use a successful multi-swarm cooperative optimizer called PS2O to minimize both the reader-to-reader interference and total system transaction time in RFID reader networks. The main idea of PS2O is to extend the single population PSO to the interacting multi-swarms model by constructing hierarchical interaction topology and enhanced dynamical update equations. As the RFID network scheduling model formulated in this work is a discrete problem, a binary version of PS2O algorithm is proposed in this study. Numerical results for four test cases with different scales, which ranging from 30 readers to 200 readers, have been presented to demonstrate the performance of the proposed methodology.

Keywords: Coevolution, PS2O, Discrete Optimization, RFID Network Scheduling

1. Introduction

In the field of optimization, nature-inspired methods have attracted more and more attentions for optimizing multimodal, nondifferentiable and discrete problems. Drew inspiration form the homogeneous cooperation within species, the most successful swarm intelligence system—Particle Swarm Optimization (PSO), has been presented [1-3]. The PSO paradigm gets a better result in a faster, cheaper way compared with other methods and has already come to be widely used in many areas [4,5]. The canonical PSO model evolves a single population (species) of interacting individuals (particles), cooperatively searching for the optimum in the D-dimensional problem space. In mathematical terms, each individual’s direction of movement is a function of its current propensity regarding the issue (the individual’s current position $x_i(t-1)$ and velocity $v_i(t-1)$), its own previous experience (the individual’s previous best successful position $p_i$), and the successes of any population member of the neighborhood (the best position found by its neighbors $p_g$):

$$x_i(t) = f(x_i(t-1), v_i(t-1), p_i, p_g)$$

(1)

However, like the previous EAs (evolutionary algorithms which drew inspiration from evolution by nature selection, such as genetic algorithm, evolutionary programming, evolution strategies and genetic programming), PSO suffer from the following drawback: as a PSO population evolves, all individuals suffer premature convergence to the local optimum in the first generations that lead to low population diversity with a adaptation stagnation as an overall result in successive generations. This loss of population (species) diversity is not observed in the natural systems. Because populations of species interact with one another in nature, they form biological communities which are large social systems typically consist of both heterogeneous and homogeneous aspects. The number of interacting species in these communities and the complexity of their relationships exemplify what is meant by the term “biodiversity”.
The Particle Swarms Swarm Optimization (PS²O) algorithm proposed here is based on the cooperative coevolution theory [6,7]. As in nature, individuals interact constantly. Within a species or a population, individual species members use information of other members to find more food more quickly and allocate more time to feed but less to look for predators than individuals do. Two or more individuals from different species or populations can also interact with each other to gain food, protection from enemies, a nesting site, or a combination of benefits. In biology, such heterogeneous cooperation and homogeneous cooperation in an ongoing cycle of adaptation are called symbiotic coevolution. Logically, researchers in the fields of evolutionary computation have modeled coevolution as optimization process.

In this paper, we implement an entire social system which consists of both interspecific cooperation and intraspecific cooperation aspects in formulating our coevolutionary simulation models. We introduced a number of \( N \) species each possesses a number of \( M \) individuals into this coevolution model to represents the “biological community”. The coevolution process in our model is hierarchical and contains three levels (i.e. individual level, species level and community level). Each individual of the biological community evolves based on the knowledge integration of itself (individual-level evolution, associate with individual’s own cognition), its species members (species-level coevolution, associate cooperative interaction within species) and its symbiotic partners from other species (community-level coevolution, associate heterogeneous cooperation between individual from different species). Clearly we model more details of the social behaviors in nature ecosystems and tie this model closer to natural evolution. Since the community is made up of a swarm of agents who are species while each species is made up of a swarm of species member, our swarms within swarm model is instantiated as a cooperative coevolutionary optimization algorithm, namely particle swarms swarm optimizer (PS²O) to solve discrete problems. In order to evaluate the performance of PS²O, extensive studies based on four real-world RFID network cases, which focusing on minimizing both the reader-to-reader interference and total system transaction time of large-scale RFID network, have been carried out. The simulation results, which are compared to other methods, are reported in this paper to show the merits of the proposed algorithm.

The paper is organized as follows. Section 2 describes the PS²O model. In Section 3, the detailed procedure of RFID network scheduling based on PS²O model are presented. The simulation results of PS²O on the RFID network scheduling problem are presented in Section 4. Section 5 concludes the paper.

2. The PS²O model

2.1. The cooperative coevolution model

In this section, we describe our model for the coevolution of symbiotic species and formulate it as an optimization algorithm. We present the outline of our model by making the following assumptions:

a) All species feel the same external environmental stress.

b) Between species, symbiotic partners cooperate with each other and all partners gain an advantage to increase their survival ability.

c) Within species, species members cooperate with each other and rely on the presence of other members for survival.

d) Cooperation both within and between species are obligate through the whole life cycles of all species.

These assumptions yield a model that can be instantiated as the optimization algorithm present below.

The population of the PS²O algorithm is called an community, which contains a species set \( \Omega = \{ S_1, S_2, \ldots, S_n \} \), and each species possesses a members set \( S_i = \{ x_{ij}^1, x_{ij}^2, \ldots, x_{ij}^M \} \). I.e., totally \( n^M \) individuals coevolve in the discrete landscape. The \( i \)th member of the \( k \)th species is characterized by the discrete vector \( x_{ik}^k = (x_{i1}^k, x_{i2}^k, \ldots, x_{im}^k) \), \( x_{ij}^k \in \{0,1\} \). In each generation \( t \), each individual \( x_{ij}^k \) behaves as follow:

1) Social evolution: this process addresses the cooperation between individuals of the same species. This species-level cooperation addresses the individual-level representation as follow: within the
species $k$, one or more neighbors of $x_i^k$ contribute their knowledge to $x_i^k$ and $x_i^k$ also share its knowledge with its neighbors. Then $x_i^k$ accelerate towards the personal best position and the best position found by its neighbors:

$$
\alpha_i^k = c_1 r_1 (p_i^k - x_i^k) + c_2 r_2 (p_g^k - x_i^k)
$$

(2)

Where $\alpha_i^k$ is the social acceleration vector of $x_i^k$, $p_i^k$ is the personal best position found so far by $x_i^k$, $p_g^k$ is the best position found so far by its neighbors within species $k$, $c_1$ are individual learning rates, $c_2$ are social learning rate; and $r_1, r_2 \in \mathbb{R}^d$ are two random vectors uniformly distributed in [0, 1].

2) Symbiotic evolution: this process addresses the community-level cooperation between distinct symbiotic species. From the perspective on community, the species cooperatively interacts with and rewards all its symbiotic partners in the biological community. This community-level cooperation addresses the individual-level representation as follow: all the members constituted the species $k$ accelerate towards the best position that the symbiotic partners of species $k$ have found. Then $x_i^k$ is manipulated according to:

$$
\beta_i^k = c_3 r_3 (p_i^k - x_i^k),
$$

(3)

Where $\beta_i^k$ is the symbiotic acceleration vector of $x_i^k$, $l$ is the index of the most successful species in the community, $c_3$ is the “symbiotic learning rate”, $r_3 \in \mathbb{R}^d$ is a uniform random vector in the range [0, 1], and $p_i^l$ is the best position found so far by the species $l$. Then the acceleration, the velocity and the position of $x_i^k$ are updated according to:

$$
\alpha_i^k = \alpha_i^k + \beta_i^k, \quad v_i^k = v_i^k + \alpha_i^k, \quad u = x_i^k + v_i^k
$$

(4)

$$
\text{if } (\text{rand}() < \text{trans}(u)), x_i^k = 1; \text{else } x_i^k = 0
$$

(5)

Here $\text{trans}(u)$ is the transfer function. In this paper, we use sigmoid function to discrete the position.

**Table 1. Pseudocode of PS2O**

<table>
<thead>
<tr>
<th>Set $t := 0$;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>INITIALIZE.</strong> Randomize positions and velocities of $N \times M$ particles in search space. Randomly divide the whole population into $N$ species each possesses $M$ particles;</td>
</tr>
<tr>
<td><strong>WHILE</strong> (the termination conditions are not met)</td>
</tr>
<tr>
<td><strong>FOR</strong> (each species $k$) <strong>IN</strong> PARALLEL</td>
</tr>
<tr>
<td><strong>FOR</strong> (each particle $i$ of species $k$)</td>
</tr>
<tr>
<td>Update the velocity and position using equations (3) and (4)</td>
</tr>
<tr>
<td><strong>END FOR</strong></td>
</tr>
<tr>
<td><strong>END FOR IN PARALLEL</strong></td>
</tr>
<tr>
<td>Set $t := t + 1$;</td>
</tr>
<tr>
<td><strong>END WHILE</strong></td>
</tr>
</tbody>
</table>
Clearly, our model for the coevolution of symbiotic species is inherently different from past ones in the following ways: 1) this model contains a number $N$ of species, and each species possesses a certain number $M$ of individuals; 2) all species are separated breeding population and concurrently search the problem space to obtain parallelism; 3) the evolution of each species is handled by an force generation equation which simulated cooperation not only within species but also between species; 4) the symbiotic relationship is inherently set between species, and all species interact one other in each generation. Cooperation is now conducting both within species and between species over the entire life cycle of the ecosystem. The pseudocode for the PS2O algorithm is listed in Table 1.

2.2. Interaction topology

It should be noted that the interaction of particles occurred in a two-level hierarchical topology in the proposed model. Many patterns of connection can be used in different levels of our model. The most common ones are rings, two-dimensional and three-dimensional lattices, stars, and hypercubes [8]. In this work, we employed the ring and the star topologies in different levels and obtained four hierarchical interaction topologies, which are illustrated in Fig. 1. In Fig. 1 (a) and (b), both levels are structured as stars (or rings). In Fig. 1 (c) and (d), four swarms at the upper level are connected by a ring (or a star), while each swarm (possesses four individual particles at the lower level) is structured as a star (or a ring). Namely, the first two topologies are characterized by homogeneous structure (employ the same patterns in both levels) and the other two have the heterogeneous structures (employ different patterns in different levels).

3. RFID network scheduling based on PS2O

3.1. RFID network scheduling problem formulation

Interferences in RFID system are usually classified into two categories [9]:
- Reader to reader frequency interference, occurs when readers are interfered with others from communicating with tags.
- Reader-to-tag interference, occurs when two or more readers in the transmission zone attempts to communicate with one tag simultaneously.
That is, given a RFID network laid out in some manner, we can construct the associated collision graph $G=(V, E)$ where each vertex $v \in V$ corresponds to a RFID reader and each edge $e \in E$ indicates that those two readers can be operated in parallel (i.e., there are no collisions between these two readers).

It should be noted that the graph partition problem (GPP) is to partition a graph $G$ into $k$ subgraphs such that the number of edges connecting nodes in different subgraphs is minimized, and the number of edges connecting nodes of the same subgraph is maximized. The frequency allocation problem for networks of RFID readers is to allocate frequencies to various readers. I.e., when two readers lie in each other's interference region, they are given different frequencies. Clearly this problem can be reduces to GPP model.

In our RFID network scheduling model, there are two objectives, namely finding the optimal transmissions of readers and scheduling the readers to reduce the total system transaction time. The first objective is in order to avoid reader interference or collision. At this point, this problem looks like the frequency allocation problem, except that the allocation is done along the time axis. Interfering readers are allotted non-overlapping periods of time so as to avoid collision between them. Obviously, the second objective is in order to confirm the efficiency of the RFID system by minimizing the total transaction time of the RFID networks. Then in each time step, the RFID network scheduling model can be formulated by four rules as follows:

1. Always maximize the number of readers in a partition and this reduces the total number of time steps after scheduling the whole RFID network.
2. A weighting value is assigned to each partition to minimize the variation in reader transaction time within the time step. This weight is defined as the average of the readers' transaction times in the partition. Besides minimizing the range of transaction values in the partition, this also retains readers with lower transaction times only for further partitioning and reduces total transaction time of the RFID system.
3. It should be noted that the higher the number of edges in the graph, the greater the possibility of finding the optimal partition. Since cut the reader with the most conflicts will cause minimum affects on the number of edges in the graph, a higher preference is given to the reader with the most conflicts.
4. The process that removing partition from the collision graph should repeat until all the readers are grouped in a number of time steps.

Following the assumption above, for each time step $t$, the solution variable is therefore given by $s' = (s'_1, s'_2, ..., s'_{n(t)})$ as a partition of the RFID network. Here $n(t)$ denotes the number of RFID readers that waiting for scheduling in this time step.

Each element $s'_i = [0, 1], i=1, 2, ..., n(t)$, in the solution vector is corresponding to the presence or absence of the $i$th reader. That is, a bit “0” in a solution vector indicated the absence of the corresponding reader, while a bit “1” means the reader’s presence. Then in each time step $t$, the RFID network scheduling model can be formulated as a discrete optimization problem that is given in what follows:

$$f(s') = w_1 f_u(s') + w_2 f_p(s') + w_3 f_e(s') + w_4 f_v(s') - \eta f_p(s')$$

(6)

where $w_1, w_2, w_3, w_4$ are the weights given to each term of the fitness function and $\eta$ is the punishment coefficient; $f_u$ is the function to select the max transaction time of the solution partition; $f_p$ is the function to calculate the number of readers in the time step of this solution; $f_e$ is the sum function of all the possible collision which the members of the partition have with the readers still remaining to be partitioned in the RFID network. Intuitively, this makes sense since removal of the partition leaves a lot of scope for further formation of partitions in remaining nodes and would not cause much loss of edges in the graph (i.e. the RFID network); $f_v$ is the weight function attached to this partition of readers that is assigned to the average of the transaction times of the RFID readers forming the partition; $f_p$ is the punish function that enable the searching algorithm exclude the illegitimate partitions that contain colliding readers.
3.2. RFID network scheduling procedure

The detailed design of RFID network scheduling algorithm based on PS²O is introduced in this section. The algorithm design reflects a multi-phase searching process as Fig.2 illustrates. The overall searching process can be described as follows:

(1) Initialization Phase

a. Reader Specification

This gives the details of the RFID readers that include 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; the process time; and the number of the 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 RFID reader network distribution (i.e., the reader position) in the working area; then the collision graph according to the readers’ layout, the interrogation and interference range, etc..

c. Population Generation

$M \times N$ individuals forming the PS²O population should be randomly generated. The $i^{th}$ particle of the $t^{th}$ time step is defined as follows:

$$X_i^t = (x_{i1}^t, x_{i2}^t, \ldots, x_{in(t)}^t), \quad x_{ij}^t \in [0,1]$$

In our work the direct encoding scheme is applied to encode the individuals. The possible solutions are represented as an particle with dynamic dimension according to different time steps (i.e., solution vector dimension $n(t)$ = the number of readers in $i^{th}$ time step). Each element $x_{ij}^t$ in the dimension is corresponding to the absence of the $j^{th}$ reader that can only be “0” or “1”.

Figure 2. The RFID network scheduling flowchart
(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 scheduling solution. The basic building blocks of this phase are:

a. Fitness Evaluation

At each iteration of every time step, for each individual \( x_i \), evaluate its fitness using the objective optimization function as follows:

\[
 f(x) = w_1 \max P(x) + w_2 \sum_{j=1}^{40} x_j' + w_3 \sum_{j=1}^{40} C(x') = 1 + w_4 \sum_{j=1}^{40} P(x') \sum_{j=1}^{40} x_j' - \eta \sum_{j=1}^{40} x_j' \tag{8}
\]

where \( P(\bullet) \) is the readers’ processing time array of the particle in time step \( t \), \( C(\bullet) \) is the sum function of all the possible collision that the reader group in individual \( x_i \) has with the readers still remaining to be partitioned in the RFID network, the fourth term in the equation is the average weight of the transaction times of the readers forming the clique in each particle, the punishment coefficient \( \eta \) is set to be 1000, and \( w_1 \gg w_2 \gg w_3 \gg w_4 \) representing the importance of each term.

b. Population Evolution

Compare the evaluated fitness values and select individual best, swarm best, and population best position for each particle in each swarm in the whole population. Then update the velocity and position of each particle according to Eqn. (2-5).
Table 2. Scheduling results on time steps of four RFID networks

<table>
<thead>
<tr>
<th>Readers</th>
<th>PS2O</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best fitness</td>
<td>Mean fitness</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>30</td>
<td>7</td>
<td>8.6333</td>
<td>0.7915</td>
</tr>
<tr>
<td>60</td>
<td>12</td>
<td>12.2333</td>
<td>1.0034</td>
</tr>
<tr>
<td>120</td>
<td>23</td>
<td>22.0633</td>
<td>1.0981</td>
</tr>
<tr>
<td>200</td>
<td>37</td>
<td>38.6000</td>
<td>1.6955</td>
</tr>
</tbody>
</table>

Table 3. Scheduling results on total processing time of four RFID networks

<table>
<thead>
<tr>
<th>Readers</th>
<th>PS2O</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Best fitness</td>
<td>Mean fitness</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>60</td>
<td>169.8854</td>
<td>196.9614</td>
<td>8.3889</td>
</tr>
<tr>
<td>120</td>
<td>310.0944</td>
<td>331.9886</td>
<td>12.2321</td>
</tr>
</tbody>
</table>

c. Termination condition

In each time step, the PS2O algorithm is performed until the fitness is small enough, or a predetermined number of iterations are passed. Then the obtained optimal reader partition should be cut from the RFID network, and the procedure will go back to the optimization phase. The computation is repeated a certain number of time steps, until all the readers are grouped and the optimal scheduling can be obtained.

4. Simulation results

In this section, four RFID reader network cases, which including 30, 60, 120 and 200 readers respectively, are scheduled to validate the capability of the proposed method. Due to the limited space, the collision graphs and reader processing times for the other two tested cases are omitted.

In the simulation test, the proposed PS2O, GA and PSO were tested on these four RFID reader network cases. The population size for all algorithms was set at 60. The max generation of each run is 300, 800, 3000, and 8000 for 30, 60, 120, and 200 readers’ network respectively. The maximum velocity was set to be 4 for both PSO and PS2O. The learning rate were set to the values $c_1 = c_2 = c_3 = 2.0$ and $c_1 = c_2 = 2.0$ for PS2O and PSO respectively. For PS2O, the swarm number $M$ was set to be 5 and the topology was as Fig.1(c). For GA, single point crossover operation with the rate of 0.8 was employed and mutation rate was set to be 0.01 [10, 11].

The experimental results, including the best, mean and standard deviation of the total processing time and the number of time steps found in 30 runs are proposed in Table 2 and 3. The convergence curves of all algorithms on four tested cases are showed in Fig.3.

From the scheduling results, the PS2O algorithm can constantly find an optimal schedule results (i.e., less time steps and total processing time). In fact, with an increasing in the number of the readers (hence the degree of the graph), the problem of finding best solution becomes intractable. However, it can be see cleanly, our proposed method is able to find the optimal schedule results of the larger scale RFID networks robustly and consistently.

5. Conclusion

This paper is devoted to giving a new strategy for scheduling reader networks in RFID-based ubiquitous computing environment. That is, a novel optimal scheduling scheme for RFID networks using a binary version of multi-swarm coevolution algorithm called PS2O is presented in this work.

PS2O extends the single population PSO to interacting multi-swarms model by constructing hierarchical interaction topologies and enhanced dynamical update equations. With the hierarchical interaction topology, a suitable diversity in the whole population can be maintained. At the same time,
the enhanced dynamical update rule significantly speeds up the multi-swarm to converge to the global optimum.

PS\textsuperscript{2}O is then employed to solve the real-world RFID network scheduling problem. The simulation studies, which also compared to PSO and GA algorithms, show that the PS\textsuperscript{2}O obtains superior RFID network scheduling solutions than the other two methods in terms of optimization accuracy and convergence speed.

6. Acknowledgements

This work was supported in part by the Natural Science Foundation of China under Grant 61105067, 71001072, 71271140 and 61174164.

7. References