5th SASTech
Iran | Mashhad
May 12 - 17 | 2011
5th Symposium on Advances in Science & Technology

Representing an efficient way to optimize coverage of sensors in wireless sensor network

Farzaneh Sheikhnezhad fard, Member of software engineering department of Khavaran Higher-education Institute, farzaneh.shfard@gmail.com

Mahmoud Bahoosh, Graduate from Khavaran Higher-education Institute, ma.bahoosh@gmail.com

Arash Keyvan, Graduate from Khavaran Higher-education Institute, ars.keyvnan@live.com

Paper Reference Number: 0502-946
Name of the Presenter: Mahmoud Bahoosh

Abstract

A wireless sensor network (WSN) consists of several sensors in a region that is distributed geographically. Each sensor has features of wireless communication for intelligent signal processing and network communications. In this paper, for optimizing coverage of sensors, Hybridization of Cellular Automata (CA) and Particle Swarm Optimization (PSO) are used. In this model, effective parameters in velocity update and position of each particle is adjusted using PSO and CA. Comparison of results of PSO and the Cellular Particle Swarm Optimization (CPSO) algorithm, shows CPSO superior to PSO algorithm. The experiments have shown that the CPSO algorithm less likely to be trapped in local optimum.

Keywords: Cellular Particle Swarm Optimization, Particle Swarm Optimization, Cellular Automata, Wireless Sensor Network, Optimizing.

1. Introduction

Recently, there has been a great attraction of interests in researches of Wireless Sensor Networks (WSN) since it is a combination of short-distance wireless telecommunication technology, Topology Control issues, Self-organizing mechanism, Micro-embedded technology, optimized network management and other modern scientific technology, comprising a set of mutually connected main sensor nodes and sink nodes of sensor based on Ad Hoc telecommunication [1].

Due to the fact that it plays an important role to improve the coverage of WSNs, it is considered to be efficient to improve the coverage by hybridization of PSO and CA. In this paper, we use Particle Swarm Optimizer algorithm to find out the best position of sensors deployment in the WSN and then optimize the WSN by adding CA algorithm to PSO algorithm after generating a quantity of nodes to constitute WSNs at random.

Heretofore, different way to optimize coverage in WSNs is presented, that PSO and CA algorithm can be mention.

2. Cellular Automata

Cellular Automata (CA) are characterized by their discretization of space and time [2]. Typically a cellular automaton consists of a graph where each node is a finite state automaton (FSA) or cell. This graph is usually in the form of a two-dimensional lattice whose cells evolve according to a global update function applied uniformly over all the cells. As arguments, this update function takes the cell's present state and the states of the cells in its interaction neighborhood as shown in Fig 1. The interaction neighborhood is a collection of nearby cells which the update function interrogates for state information. The size and shape of neighborhoods vary from application to application. As the CA evolves, the update function will determine how microscopic, or local, interactions influence each other to the overall macroscopic behavior of a complete system [2].

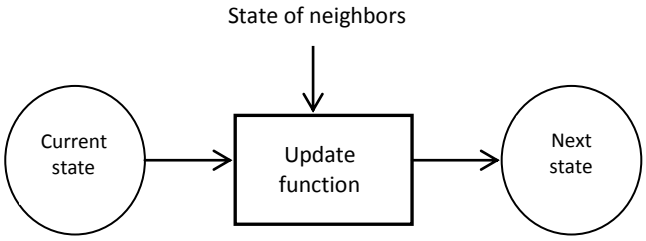

Fig 1: State transitions depend on neighborhood function[2].

3. **Particle Swarm Optimization**

Due to its simple implementation, minimum mathematical processing and good optimization capability, PSO has attracted more attentions. As Vassiliadis and Dounias summarized, PSO can be considered the most popularly and frequently applied among nature-inspired techniques according to related publications [3].

PSO simulates the behavior of a bird flock. When a flock of birds forage for food, two simple and important strategies are:

(a) Searching for the peripheral region around the bird that is nearest to the food; (b) judging the position of the food by its own flying experience. Inspired by the two strategies, the search space of optimization problem is regarded as birds' flying space; every bird is abstracted as a particle with non-quality and non-volume so as to denote a candidate solution; and the optimal solution to be searched for the problem is the food to all the particles. Then, the basic PSO algorithm comprises a swarm of particles moving in the D-dimensional search space which includes all possible candidate solutions.

We denote the ith particle as $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})$, and its flying velocity as $V_i = (v_{i,1}, v_{i,2}, \ldots, v_{i,d})$. Depending on the two strategies above, when each particle ''flies" in the search space, $P_i = (p_{i,1}, p_{i,2}, \ldots, p_{i,d})$ denotes the personal best position the ith particle has found so far, and $P_g = (p_{g,1}, p_{g,2}, \ldots, p_{g,d})$ is the global best position discovered by the swarm. At each time step t, both of each particle's velocity and position are updated so that a particle moves to a new position. The following two equations are employed to calculate the velocity and position:

$$V_i^{t+1} = V_i^t + c1r1(P_i^t - X_i^t) + c2r2(P_g^t - X_i^t) \quad (1)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2)$$

Where c1 and c2 are two positive constants (acceleration constants), r1 and r2 are two uniform random numbers in [0, 1]. A constant V_{max} is often used to limit each particle's velocity to guarantee that most of the new positions of the particles will be restricted in the search space of feasible region at each iteration. Eqs (1) and (2), lead to the movement of each particle towards its cognitive best position (P*i*) and "social" best position (P_g) with random perturbations caused by c1r1 and c2r2, respectively. The quality of a particle's position is measured by its fitness value, or namely, the better fitness value means the better position [4].

4. **Hybridization algorithm CPSO**

Let us consider that a set of cells are distributed in checkerboard-like lattice, each cell is tightly connected with its neighboring cells. Each cell has different states which could be considered as its intrinsic information. At a discrete time step, each cell communicates with its neighbors, and updates its current state according to its communication. All cells communicate and change states synchronously. According to the concept of CA, four basic components of CA could be summarized as: cell state, cell space, neighborhood, and transition rule. Cell state is the number of distinct states that a single cell can be in. A cell space describes how cells are connected with each other. In two-dimensional CA, a lattice structure is used to represent the cell space. Because we usually simulate real-world dynamic systems by finite lattice grids, we need to define the boundary condition. And we introduce periodic boundary by means of imaging the lattice grids embedded in toroidal surface topology, that is, the left boundary connects to the right boundary, and the upper boundary connects to the lower boundary. Neighborhood is a set of cells surrounding a given cell. It affects the next state of the given cell (e.g. Moore neighborhood, which is a set of cells surrounding a given cell with the radius equaling to 1). Transition rule is the control component in CA; it is used to determine the next state of a cell according to its current state and the states of the cells in its neighborhood. So CA could be described briefly as ‘‘In a preassigned cell space, every cell updates its current cell state governed by a transition rule according to the state of cells in the neighborhood at discrete time steps” [5].

Although CA and PSO come from different fields, comparing the two mechanisms, we find that they have several similar features. Firstly, both of them are comprised of a set of individuals, which are called cells in CA, and particles, in PSO. Secondly, every individual interacts with each other to transmit certain intrinsic information. In CA, each cell communicates with its neighbors and changes its cell state taking into account the cell’s current state and its neighbors’ states. Similarly, each particle communicates with other particles and updates such intrinsic information as velocity, position, fitness, personal best position, global best position in PSO. Thirdly, in CA, a transition rule is used to govern the evolution of cells, while we use two equations about velocity and position to update particles. Finally, CA and PSO both run in discrete time step[1].

PSO mimics a swarm system that involves interaction between different particles in the swarm. In order to enhance the performance of the interaction, we focus on employing the idea of CA to explore the communication structure and information inheriting and diffusing mechanisms of the swarm system of PSO. So in the proposed CPSO, we analyze PSO with a CA model, where individuals can only exchange information within their neighborhood. It could help exploring the search space due to the slow information diffusion through the population, promoting the preservation of diversity, and exploiting every cell’s local information inside the neighborhood. Note that when constructing the CA model for CPSO, instead of rigorously and rigidly combining the concepts and methods of CA and PSO, we design a mechanism sophisticatedly by integrating adapted CA and PSO so that concentration can be attained. The CA model for PSO is defined as follows:

(a) Cell: selected candidate solution (we assume there are U cells);
(b) Cell space: the set of all cells;
(c) Cell state: the particle’s information of such respects as P_i, P_g and X_i at time t, denoted by Si^t (i=1, 2, … , U). We let $S_i^t = [P_i^t, P_g^t, V_i^t, X_i^t, \ldots]$;
(d) Neighborhood: a kind of topology based on lattice structure containing a set of particles, defined by $N(i) = \{i+d_1, i+d_2, \ldots, i+d_m, \ldots, i+d_l\}$ where l is the size of the set, m = 1,2, ... ,l;
(e) Transition rule: $S_i^{t+1} = f(S_i^t, S_{i+d2}^t, S_{i+d2}^t, \ldots, S_{i+dl}^t)$
(f) Discrete time step: iterations in PSO[5].

```
Initialize swarm size and PSO parameters
  Initialize particles' positions and velocities
Loop
    Identify every particle's state
    Implement Cellular automata mechanism update
    velocities
    Update Positions
    Exit loop if criterion is met
End loop
```

Fig 2: The general framework of CPSO [5].

5. Results and Analysis

In this section, optimizing coverage in WSN using standard PSO and CPSO algorithms are compared and reviewed with different iterations that the result of comparisons can be seen in Table 1.

Iteration	PSO	CPSO
10	%41	%44
20	%43	%45
30	%45	%47
40	%46	%53
50	%47	%53
60	%50	%54
70	%51	%56
80	%51	%58
90	%51	%60
100	%51	%61

Table 1. Comparison coverage of PSO and CPSO algorithms

The important point is that every iteration of reviewed, coverage of the CPSO algorithm is more than standard PSO algorithm. It should be noted that in all iterations, the parameters c1 and c2 equal to 1.494 and the value of w equal to 0.729 is considered that these values are standards in [2]. In addition to, the neighborhood radius of each sensor (R) equal to 32, the maximum coverage radius of each sensor (CR) equal to 40.

In Fig 3, you can see mode of ordering sensors that the coverage of CPSO is 61% and coverage of standard PSO is 51%.

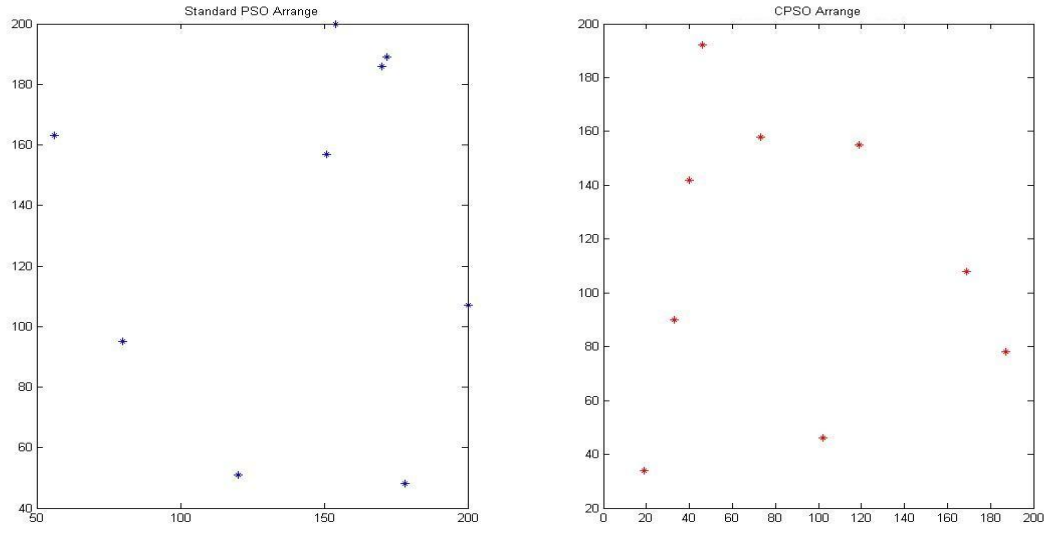

Fig 3: Mode of ordering sensors PSO and CPSO algorithms

6. Conclusions

In this paper hybridization of CA and PSO algorithms, namely CPSO, is used. In this model, effective parameters in velocity update and position of each particle is adjusted using PSO and CA.

The empirical results show that converge ratio of CPSO algorithm is almost 10% better than PSO algorithm in average. The CPSO algorithm not only augments converge, but it reduces the optimization error as well. The experiments have shown that the CPSO algorithm less likely to be trapped in local optimum.

Therefore the CPSO algorithm can move the particles faster towards global optimum with bearing less error.

7. References

[1] D. Tian, N.D. Georganas. (2002). A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks. *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, 32–41.

[2] E. F. Codd. Cellular Automata. (1967). Academic Press.

[3] J. Kennedy, R. Eberhart. (2001). *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann.

[4] Matthew Settles, Terence Soule. (2005). Breeding Swarms: A GA/PSO Hybrid. *ACM Journal.*

[5] Yang Shi, Hongcheng Liu, Liang Gao, Guohui Zhang. (2010). Cellular particle swarm optimization. *Information Sciences*, 2–5.