

Routing Based Ant Colony Optimization in Wireless Sensor Networks

Dr. Anjali¹ and Navpreet Kaur²

¹ Lovely Professional University Phagwara, India

Received: 13 December 2012 Accepted: 5 January 2013 Published: 15 January 2013

Abstract

Wireless Sensor Networks (WSN's) have become an important and challenging issue in last year. Wireless Sensor Networks consist of nodes with limited power are deployed to gather useful information from the field and send the gathered data to the users. In WSNs, it is critical to collect the information in an energy efficient manner. Ant Colony Optimization, a swarm intelligence based optimization technique, is widely used in network routing. In this paper, we introduce a heuristic way to reduce energy consumption in WSNs routing process using Ant Colony Optimization. We introduce three Ant Colony Optimization algorithms, the Ant System, Ant Colony System and improved AS and their application in WSN routing process. The simulation results show that ACO is an effective way to reduce energy consumption and maximize WSN lifetime.

Index terms— ant colony optimization, routing, wireless sensor network.

1 Introduction

ue to advance information technology, Wireless sensor networks (WSN's) are rapidly developing area in both research and application. The wireless sensor networks are based on the cooperation of a number of tiny sensors and which are depending upon the four parts: sensor (motes), processor, transceiver, and battery. The Sensor get information from surrounding area and processor change the analog information into digital information. Wireless sensors have the ability to perform simple calculations and communicate in a small area. Wireless sensor networks have critical applications in the scientific, medical, commercial, and military domains. Although WSNs are used in many applications, they have several limitations including limited energy supply and limited computation and communication abilities. These limitations should be considered when designing protocols for WSNs. In sensor networks, minimization of energy consumption is considered a major performance criterion to provide maximum network lifetime. When considering energy conservation, routing protocols should also be designed to achieve fault tolerance in communications.

There are two types of WSNs: structured and unstructured. An unstructured WSN is one that contains a dense collection of sensor nodes. The sensor nodes may be deployed in an ad hoc manner into the field. Once deployed, the network is left unattended to perform monitoring and reporting functions. In an unstructured WSN, network maintenance such as connection management and failures detection is difficult since there are so many nodes to take care of. In a structured WSN, all or some of the sensor nodes are deployed in a pre-planned manner. The advantage of a structured network is that fewer nodes can be deployed with lower network maintenance and management cost. Fewer nodes can be deployed now since nodes are placed at specific locations to provide coverage while ad hoc deployment can have uncovered regions.

The basic method to transfer information from a sensor node to the base is called flooding. The optimization of network parameters for WSN routing process to provide maximum service life of the network can be regarded as a combinatorial optimization problem. Many researchers have recently studied the collective behavior of biological species such as ants as an analogy providing a natural model for combinatorial optimization problems. Ants in a

colony are able to converge on the shortest among multiple paths connecting their nest and a food source. The driving force behind this behavior is the use of a volatile(D D D D D D D)

chemical substance called pheromone. While locating food, ants lay pheromone on the ground, and they also go in the direction where the concentration of pheromone is higher. This mechanism allows them to mark paths and subsequently guide other ants, and let good paths arise from the overall behavior of the colony.

The main goal of our study was to maintain network life time at a maximum, while discovering the shortest paths from the source nodes to the base node using swarm intelligence based optimization technique called ACO. The Ant Colony Optimization (ACO) is a family member of the Swarm Intelligence based approaches applied for optimization problems. A multipath data transfer is also accomplished to provide reliable network operations, while considering the energy levels of the nodes. Wireless Sensor Network architecture, ACO algorithm for network routing and Simulation Results are presented in the following sections.

2 II.

3 Overview of Routing Based Ant

Algorithm for WSN a) Ant Colony Optimization

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. This algorithm is a member of the ant colony algorithms family, in swarm intelligence methods, and it constitutes some metheuristic optimizations. Initially proposed by Marco Dorigo in 1992 in his PhD thesis, the first algorithm was aiming to search for an optimal path in a graph, based on the behavior of an ants seeking path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants.

A combinatorial optimization problem is a problem defined over a set $C = c_1, \dots, c_n$ of basic components. A subset S of components represents a solution of the problem; $F \subseteq 2C$ is the subset of feasible solutions, thus a solution S is feasible if and only if $S \in F$. A cost function z is defined over the solution domain, $z : 2C \rightarrow \mathbb{R}$, the objective being to find a minimum cost feasible solution S^* , i.e., to find $S^* \in F$ and $z(S^*) \leq z(S), \forall S \in F$. Given this, the functioning of an ACO algorithm can be summarized as follows:-A set of computational concurrent and asynchronous agents (a colony of ants) moves through states of the problem corresponding to partial solutions of the problem to solve. They move by applying a stochastic local decision policy based on two parameters, called trails and attractiveness.

By moving, each ant incrementally constructs a solution to the problem. When an ant completes a solution, or during the construction phase, the ant evaluates the solution and modifies the trail value on the components used in its solution. This pheromone information will direct the search of the future ants.

Furthermore, an ACO algorithm includes two more mechanisms: trail evaporation and, optionally, daemon actions. Trail evaporation decreases all trail values over time, in order to avoid unlimited accumulation of trails over some component. Daemon actions can be used to implement centralized actions which cannot be performed by single ants, such as the invocation of a local optimization procedure, or the update of global information to be used to decide whether to bias the search process from a non-local perspective.

More specifically, an ant is a simple computational agent, which iteratively constructs a solution for the instance to solve. Partial problem solutions are seen as states. At the core of the ACO algorithm lies a loop, where at each iteration, each ant moves (performs a step) from a state s to another one s' , corresponding to a more complete partial solution. That is, at each step t , each ant k computes a set $A_k(t)$ of feasible expansions to its current state, and moves to one of these in probability. The probability distribution is specified as follows. For ant k , the probability $p_{ss'}^k$ of moving from state s to state s' depends on the combination of two values:

1. The attractiveness $\eta_{ss'}$ of the move, as computed by some heuristic indicating the a priori desirability of that move; 2. The trail level $\tau_{ss'}$ of the move, indicating how proficient it has been in the past to make that particular move: it represents therefore an a posteriori indication of the desirability of that move. Trails are updated usually when all ants have completed their solution, increasing or decreasing the level of trails corresponding to moves that were part of "good" or "bad" solutions, respectively.

b) AS (Ant system algorithm)

The first ACO algorithm was called the Ant system and it was aimed to solve the travelling salesman problem, in which the goal is to find the shortest round-trip to link a series of cities. The general algorithm is relatively simple and based on a set of ants, each making one of the possible round-trips along the cities. At each stage, the ant chooses to move from one city to another according to some rules: 1. It must visit each city exactly once; 2. A distant city has less chance of being chosen (the visibility). 3. The more intense the pheromone trail laid out on an edge between two cities, the greater the probability that that edge will be chosen. 4. Having completed its journey, the ant deposits more pheromones on all edges it traversed, if the journey is short. ACS was the first algorithm inspired by real ant's behavior. The merit is used to introduce the ACO algorithms and to show the potentiality of using artificial pheromone and artificial ants to drive the search of always better solutions for complex optimization problems. In ACS once all ants have computed their tour (i.e. at the end of each iteration) AS updates the pheromone trail using all the solutions produced by the ant colony. Each edge belonging to one of the computed solutions is modified by an amount of pheromone proportional to its solution value. At the

end of this phase the pheromone of the entire system evaporates and the process of construction and update is iterated. On the contrary, in ACS only the best solution computed since the beginning of the computation is used to globally update the pheromone. As was the case in AS, global updating is intended to increase the attractiveness of promising route but ACS mechanism is more effective since it avoids long convergence time by directly concentrate the search in a neighborhoods of the best tour found up to the current iteration of the algorithm.

4 Global

ANTS algorithm within the ACO frame-work has two mechanisms:

5 i. Attractiveness

The attractiveness of a move can be effectively estimated by means of lower bounds (upper bounds in the case of maximization problems) on the cost of the completion of a partial solution. In fact, if a state s corresponds to a partial problem solution it is possible to compute a lower bound on the cost of a complete solution containing s .

6 ii. Trail Update

A good trail updating mechanism avoids stagnation, the undesirable situation in which all ants repeatedly construct the same solutions making any further exploration in the search process impossible. Stagnation derives from an excessive trail level on the moves of one solution, and can be observed in advanced phases of the search process, if parameters are not well tuned to the problem. The trail updating procedure evaluates each solution against the last k solutions globally constructed by ANTS. As soon as k solutions are available, their moving average z is computed; each new solution z is compared with z . If z is lower than z , the trail level of the last solution's moves is increased, otherwise it is decreased. $\hat{\tau}_{i,j} = \tau_{i,j} \cdot (1 - z_{curr} - LB/z - LB)$

Where z is the average of the last k solutions and LB is a lower bound on the optimal problem solution cost.

7 III.

8 ACO Approach

In the ACO based approach, each ant tries to find a path in the network, providing minimum cost. Ants are launched from a source node s and move through neighbor repeater nodes r_i , and reach a final destination node (sink) d . Whenever, a node has data to be transferred to the destination which is described as a base or base station, launching of the ants is performed.

After launching, the choice of the next node r is made according to a probabilistic decision rule (1):

$$(1)$$

Where $\tau(r,s)$ is the pheromone value, $\eta(r,s)$ is value of the heuristic related to energy, R_s is the receiver nodes. For node r , $tabu_r$ is the list of identities of received data packages previously. α and β are two parameters that control the relative weight of the pheromone trail and heuristic value. Pheromone trails are connected to arcs. Each arc (r,s) has a trail value. $\tau(r,s) \in [0,1]$; Since the destination d is a stable base station, the last node of the path is the same for each ant travel. The heuristic value of the node r is expressed by equation (2): $(\eta(r,s) = \frac{1}{D(r,s)})$

Where I is the initial energy, and e_r is the current energy level of receiver node r . This enables decision making according to neighbor nodes' energy levels, meaning that if a node has a lower energy source then it has lower probability to be chosen. Nodes inform their neighbors about their energy levels when they sense any change in their energy levels.

In traditional ACO, a special memory named M_k is held in the memory of an ant to retain the places visited by that ant (which represent nodes in WSNs). In equation (3), the identities of ants (as sequence numbers) that visited the node previously, are kept in the node's memories, instead of keeping node identities in ant's memories, so there is no need to carry M_k lists in packets during transmission. This approach decreases the size of the data to be transmitted and saves energy.

In equation (3) each receiver node decides whether to accept the upcoming packet of ant k or not, by checking its $tabu$ list. So, the receiver node r has a choice about completing the receiving process by listening and buffering the entire packet. If the receiver node has received the packet earlier, it informs the transmitter node by issuing an ignore message, and switches itself to idle mode until a new packet arrives.

After all ants have completed their tour, each ant k deposits a quantity of pheromone $\hat{\tau}_{i,j}^{k(t)}$ given in equation (3), where L_k is the length of tour, $w_k(t)$ is done by ant k at iteration t . The amount of pheromone at each connection $((l(r,s))$ of the nodes is given in equation (4). In WSNs, N represents the total number of nodes visited by ant k of tour w at iteration t :

$$(3)$$

Pheromone values are stored in a node's memory. Each node has information about the amount of pheromone on the paths to their neighbor nodes. After each tour, an amount of pheromone trail $\hat{\tau}_{i,j}^{k(t)}$ is added to the path visited by ant k . This amount is the same for each arc (r, s) visited on this path. This task is performed by sending ant k back to its source node from the base along the same path, while transferring an acknowledgement

signal for the associated data package. Increasing pheromone amounts on the paths according to lengths of tours, $J_w(t)$ would continuously cause an increasing positive feedback. In order to control the operation, a negative feedback, the operation of pheromone evaporation after the tour is also accomplished in equation (5). A control coefficient $\alpha \in (0, 1)$ is used to determine the weight of evaporation for each tour [19]: (5) In simulations, ACO parameter settings are set to values 2 for α , 6 for β , and 0.5 for ρ , which were experimentally found to be good by [20].

IV.

9 Simulation

In this section, we present the performance results of the simulation experiments.

10 Conclusion

In this paper, Ant Colony Optimization based routing algorithm was implemented. In WSN, the life time network is depended essentially to the density and the rate of communications of sensors which affect the battery level and so the network. We act on the routing level and present a new routing algorithm, which uses ant colony optimization algorithm for WSNs. This solution improves actively the life time network of the WSN. The next work will be focused on the mobility context of sensors which considered as a huge challenge in WSN area with energy consumption metric.

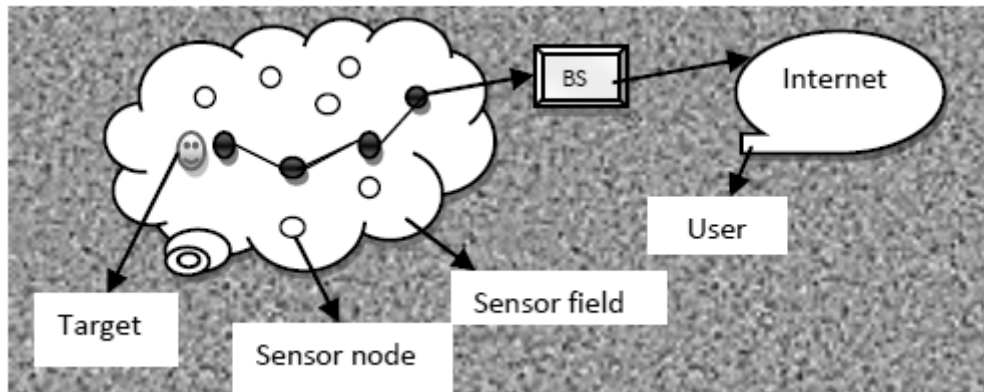


Figure 1: Figure 1 :

¹© 2013 Global Journals Inc. (US) Year

²© 2013 Global Journals Inc. (US)

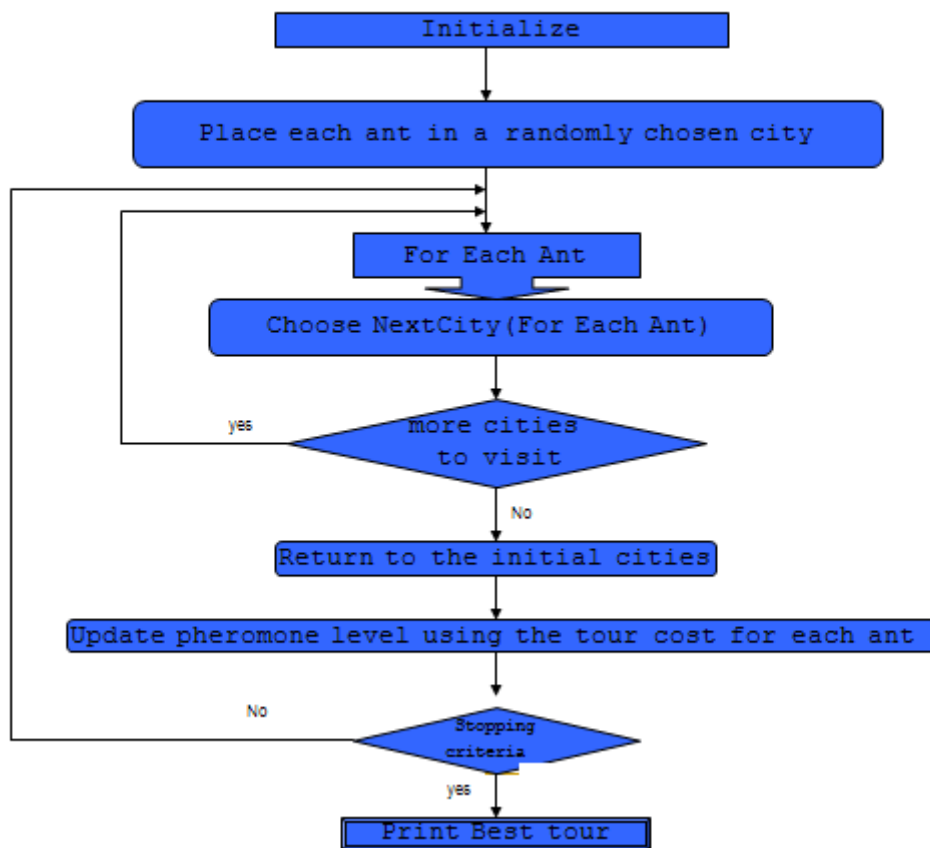
³© 2013 Global Journals Inc. (US) Year



5

Figure 2: 5 .

Ant Systems Algorithm



2

Figure 3: Figure 2 :

$$P_k(r,s)=\begin{cases} \frac{[\tau(r,s)]^\alpha \cdot [\eta(r,s)]^\beta}{\sum_{r \in R_S} [\tau(r,s)]^\alpha \cdot [\eta(r,s)]^\beta} & \text{if } k \notin \text{tabu}^r \\ 0 & \text{otherwise} \end{cases}$$

Figure 4:

$$\eta(r,s)=\frac{(I-e_r)^{-1}}{\sum_{n \in R_S} (I-e_n)^{-1}}$$

3

Figure 5: Figure 3 :E

$$J^k_w(t)$$

345

Figure 6: Figure 3 :Figure 4 :Figure 5 :

.1 Acknowledgement

The authors also would like to thank the anonymous reviewers for their comments and suggestions to improve this paper. The authors also would to thanks my friends.

[Camara and Loureiro ()] ‘A novel routing algorithm for ad hoc networks’. D Camara , Aaf Loureiro . *System Sciences*, 2000.

[Dorigo and Bimttari] *Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique*, Marco Dorigo , Mauro Bimttari , Thomas . IRIDIA - TECHNICAL REPORT SERIES: TRIRIDIAI2006-023.

[Caro et al. ()] Gianni Di Caro , Frederick Ducatelle , Luca Maria Gambardella . *SWARM INTELLIGENCE FOR ROUTING IN MOBILE AD HOC ETWORKS*, 2005. IEEE.

[Heo et al. (2009)] ‘EARQ: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks’. J.-Y Heo , J.-M Hong , Y.-K Cho . *IEEE Trans. Ind. Informat* Feb. 2009. 5 (1) p. .

[Golshahi et al. ()] ‘Implementing An ACO Routing Algorithm For AD-HOC Networkcs’. M Golshahi , M Mosleh , M Kheyrandish . *IEEE, International Conference on Advanced Computer Theory and Engineering*, 2008.

[Bonabeau and Dorigo ()] *Natural to Artificial Systems*, E Bonabeau , M Dorigo . 1999. London, U.K.: Oxford Univ. Press. p. . (G.T. Swarm Intelligence)

[White and Pagurek ()] ‘Toward Multi-Swarm Problem Solving in Networks’. T White , B Pagurek . *International Conference on Multi Agent Systems*, 1998. p. .