Energy Efficient Routing in Wireless Sensor Networks based on Ant Colony Optimization

Okafor, Friday Onyema[1] and Fagbohunmi, Griffin Siji[2]

[1]Department Of Computer Science,
Michael Okpara University Of Agriculture, Umudike
E_Mail: Okafor.Friday@Mouau.Edu.Ng
[2]Department Of Computer Science, Abia State Polytechnic, Aba
E_Mail: Oluwasijibomi1@Hotmail.Com

ABSTRACT

Wireless Sensor Networks (WSN's) have become an important and challenging research area in recent years. Wireless Sensor Networks consisting of nodes with limited power are deployed to gather useful information from the field. In WSNs it is critical to collect the information in an energy efficient manner. Ant Colony Optimization, a swarm intelligence based optimization technique, has been successfully 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 Ant System and their application in WSN routing process. The simulation results show that Ant Colony Optimization is an effective way to reduce energy consumption and maximize WSN lifetime.

Keywords:- Ant colony optimization , Delay , Energy consumption, Routing , Routing protocols, Sensors , Wireless sensor networks.

1.0 INTRODUCTION

Due to the advancement in information technology, Wireless sensor networks (WSN's) are rapidly developing area in both research and application. Wireless sensors have the ability to perform simple calculations and communicate within the shortest possible time 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 restrictions including limited energy supply and limited computation and communication abilities. These limitations should be considered when designing protocols for WSNs. Because of these considerations specific to WSNs, many routing schemes using end-to-end devices and Mobile ad-hoc networks (MANETs) are inappropriate for WSNs [2]. 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. In addition, since channel bandwidth is limited, protocols should have capability of performing local collaboration to reduce bandwidth M requirements [4]

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 behaviour 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 behaviour is the use of a volatile 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 highest. This mechanism allows them to mark paths and subsequently guide other ants, and let good paths arise from the overall behaviour 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 Ant Colony Optimization (ACO). A multi-path data transfer is also accomplished to provide reliable network operations, while considering the energy levels of the nodes.

2.0 Overview of Routing Based Ant Algorithm for Wsn

2.1 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 meta-heuristic optimizations [6], Mauro Bimttari [6]. 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 behaviour of ants seeking a 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 behaviour of ants.

A combinatorial optimization problem is a problem defined over a set $\mathbf{C} = \{c_1, \ldots , c_n \}$ of basic components. A subset **S** of components represents a solution of the problem; $\mathbf{F} \subseteq \mathbf{C}$ is the subset of feasible solutions, thus a solution **S** is feasible if and only if $\mathbf{S} \in \mathbf{F}$. A cost function z is defined over the solution domain, $z : \mathbf{C} \in \mathbf{R}$, the objective being to find a minimum cost feasible solution S*, i.e., to find S*: $S^* \in \mathbf{F}$ and $z(S^*) \leq z(\mathbf{S})$, $\forall \mathbf{S} \in \mathbf{F}$. 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. During the construction phase or the completion of a solution, 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 to another one ψ, corresponding to a more complete partial solution. That is, at each step σ, each ant k computes a set **A(**k σ(ι)) 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(ιψ$_k$) of moving from state ι to state ψ depends on the combination of two values:

• the attractiveness η(ιψ) of the move, as computed by some heuristic indicating the a priori desirability of that move;

• the trail level τ(ιψ) 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.

2.2 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 the following 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;
5. After each iteration, trails of pheromones evaporate.

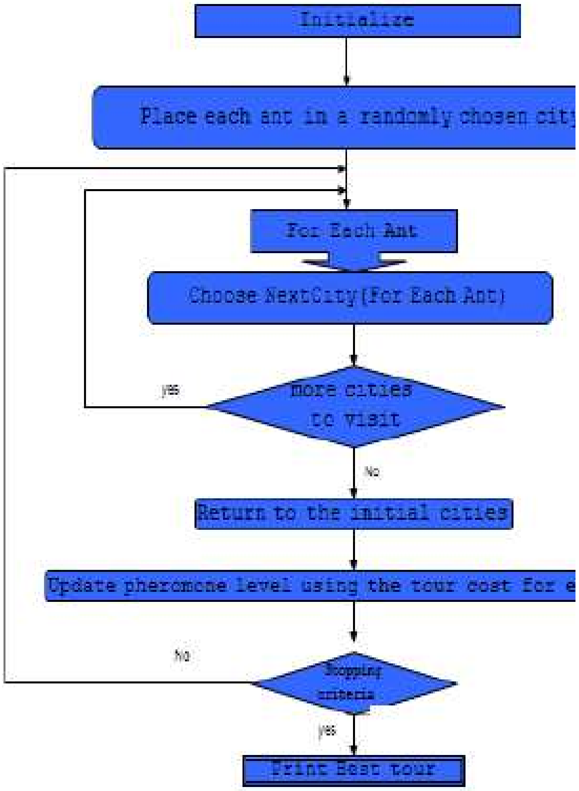
Initialize
Place each ant in a randomly chosen city
For Each Ant
Choose NextCity(For Each Ant)
yes
more cities to visit
No
Return to the initial cities
Update pheromone level using the tour cost for e
No
Stopping criteria
yes
Print Best tour

Fig. 1. Ant System Algorithm

2.3 ACS (Ant Colony System)

ACS was the first algorithm inspired by real ant's behaviour. 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 (Golshahi M, et al 2008). 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 the promising route but ACS mechanism is more effective since it avoids long convergence time by directly concentrating the search in a neighborhoods of the best tour found up to the current iteration of the algorithm [1].

ANTS algorithm within the ACO framework has two mechanisms:

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 ι corresponds to a partial problem solution it is possible to compute a lower bound on the cost of a complete solution containing ι.

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_i is compared with z. If z_i is lower than z, the trail level of the last solution's moves is increased, otherwise it is decreased.

Δτi,j = τ0 . (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.

3.0 Improved 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 neighbour 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 [5]. After launching, the choice of the next node *r* is made according to a probabilistic decision rule (1):

$$P_k(r,s)=\begin{cases}\dfrac{[\tau(r,s)]^{\alpha}\cdot[\eta(r,s)]^{\beta}}{\sum\limits_{r\in R_s}[\tau(r,s)]^{\alpha}\cdot[\eta(r,s)]^{\beta}} \text{ if } k\notin tabu^r \\ 0 \qquad\qquad\qquad\qquad \text{otherwise}\end{cases} \quad (1)$$

where *τ* (*r*,*s*) is the pheromone value, *n* (*r*,*s*) is the value of the heuristic related to energy, *Rs* in 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. 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) shown below:

$$\eta(r,s)=\frac{(I-e_r)^{-1}}{\sum_{n\in R_s}(I-e_n)^{-1}} \qquad (2)$$

where *I* is the initial energy, and e_r is the current energy level of receiver node *r [7]*. 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 neighbours 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 (1), 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 (1) 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 $\Delta\tau^k(t)$ given in equation (3), where $J_w^k(t)$ is the length of tour, w^k (t) which 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, $J_w^k(t)$ represents the total number of nodes visited by ant k of tour w at iteration t:

$$\Delta\tau^k(t)=1 / J_w^{\mathrm{k}}(t) \qquad (3)$$

$$\tau(r, s)(t) \leftarrow \tau(r, s)(t) + \Delta\tau(r, s)(t), \ \forall\ l(r, s) \in$$

$$wk(t),\ k = 1, , m \qquad (4)$$

Pheromone values are stored in a node's memory. Each node has information about the amount of pheromone on the paths to their neighbour nodes. After each tour, an amount of pheromone trail $\Delta\tau^k$ 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, Jw (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 $\rho \in (0,1)$
is used to determine the weight of evaporation for each tour

$$\tau ij(t) \leftarrow (1 - \rho)\tau ij(t) \qquad (5)$$

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 Dorigo. .

4.0 Simulation

In this section, we present the performance results of the simulation experiments. To accomplish the experiments, a parallel discrete event-based platform was developed in MATLAB. The fig. 2. are represent sensor node are randomly deploy in the network and connect with each other. Fig. 3 represent the node are randomly deployed and node are in range of each other. The nodes are connected within radius of each other. Fig.4. represent all the node is communicating in range of each other and data can be transfer in small area. The Fig. 5 represents ant movement in network to find optimize path. All Sensor nodes are communicating bidirectional and data can be transfer through ants.

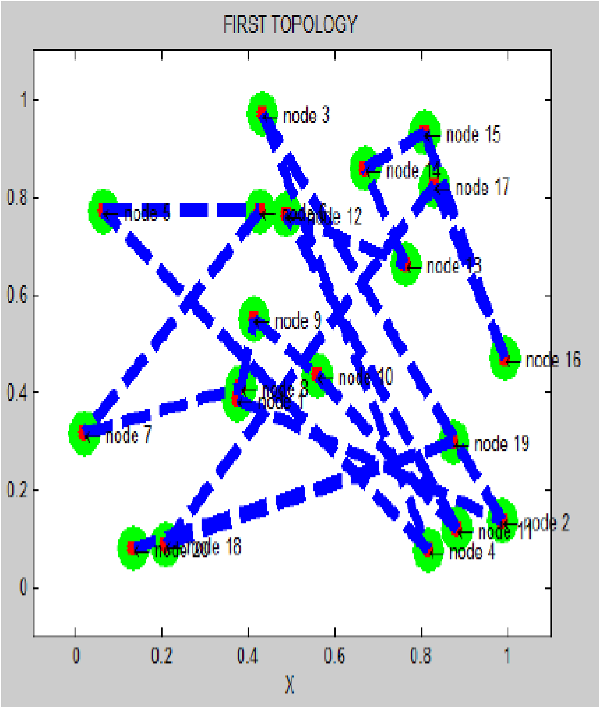

Fig. 2. Node are deployed randomly

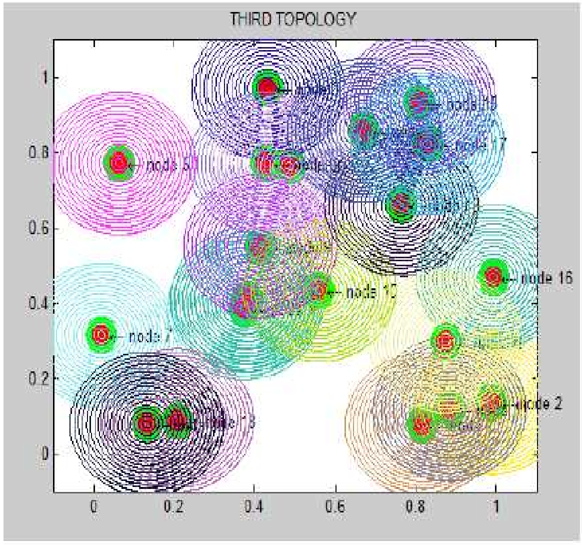

Fig. 4. All node are communicating in range of each

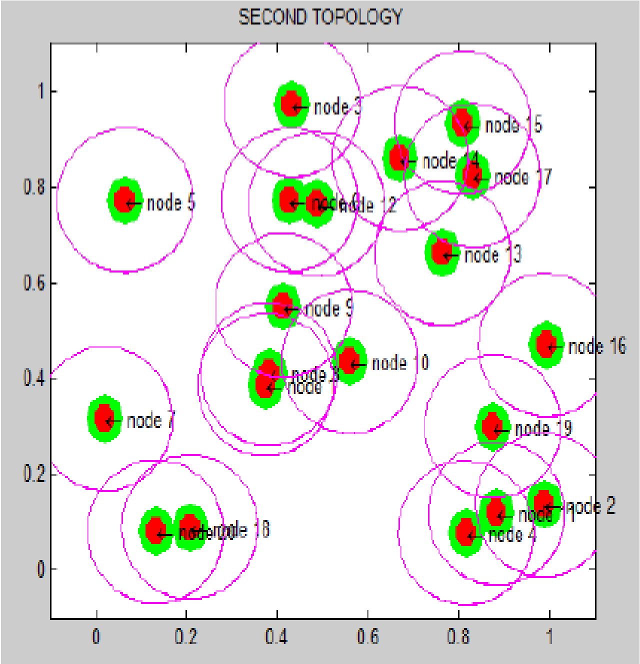

Fig. 3. All sensor node are in range of each other

In Fig. 4, all nodes are communicating in range of each other

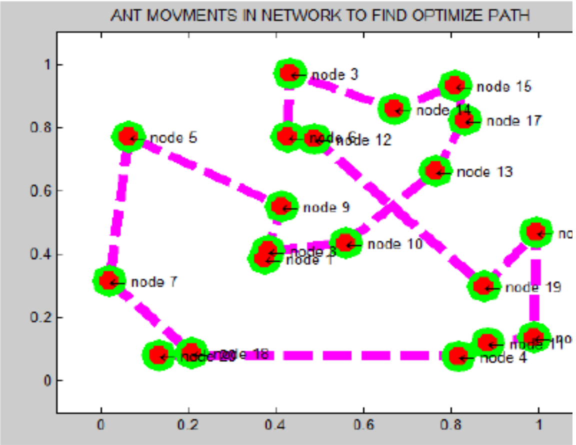

Fig.5. Ant movement in network to find optimize path

5.0 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 is considered as a huge challenge in WSN area with energy consumption metric

.

References

[1] Bonabeau E., Dorigo M.(1999) Natural to Artificial Systems. Oxford Univ. Press; London, U.K.: 1999. G.T. Swarm Intelligence; pp. 199 –278.

[2] Camara, D. Loureiro, A, (2000) "A novel routing algorithm for ad hoc networks", System Sciences journal Vol 2 pp 35 – 40.

[3] Gianni Di Caro, Frederick Ducatelle and Luca Maria Gambardella,(2005) "SWARM INTELLIGENCE FOR ROUTING IN MOBILE AD HOC ETWORKS" .IEEE Journal on Machine learning Vol 4 pp 123 – 134.

[4] Golshahi M, M.Mosleh, M.Kheyrandish,(2008) "Implementing An ACO Routing Algorithm For AD-HOC Networlcs," IEEE, International Conference on Advanced Computer Theory and Engineering. Vol 1 pp 46 - 58

[5] Heo J, J.Hong M, and Cho Y (2009), "EARQ: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks," IEEE Trans. Ind. Informat, vol. 5, no. 1, pp. 3–11.

[6] Marco Dorigo, Mauro Bimttari,(2006), "Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique", IRIDIA – TECHNICAL REPORT SERIES.

[7] White T., Pagurek B.(2009) Toward Multi-Swarm Problem Solving in Networks.IEEE journal on swarm intelligence Vol3 pp 87 - 99