An Efficient Ant Colony Algorithm for Node Deployment in Wireless Sensor Networks

Tehreem Qasim

Department of Electronics

Quaid-i-Azam University, Islamabad
Pakistan

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of

Master of Philosophy

in

Electronics

February, 2015

DEPARTMENT OF ELECTRONICS
QUAID-I-AZAM UNIVERSITY
ISLAMABAD

A thesis entitled "An Efficient Ant Colony Algorithm for Node Deployment in Wireless Sensor Networks" by Tehreem Qasim in partial fulfilment of the requirements for the degree of Master of Philosophy, has been approved and accepted by the following

---

Advisor
Dr. Hasan Mahmood
Assistant Professor
Department of Electronics
Quaid-i-Azam University, Islamabad
Pakistan

Chairman
Dr. Farhan Saif
Department of Electronics
Quaid-i-Azam University, Islamabad
Pakistan

Dedicated to my family for their unconditional love and support.

Acknowledgements

First of all I would like to thank my supervisor Dr. Hasan Mahmood for his kind support and guidance throughout my research work. His friendly demeanor and encouragement enabled me to achieve my goals.

I would also like to thank other faculty members including the chairman of department Dr. Farhan Saif, Dr. Qaisar Abbas Naqvi, Dr. Aqeel Abbas, Dr. M. Naeem A. Bhatti and Dr. Muhammad Zia who contributed a lot to enhance my knowledge. They will always be my role models.

I am grateful to my group member Asad Saleem, my sweet friends Muhamad Usman, Muhamad Kamran, Muhamad Yasir, Abdus Salam, Bilal, Faizan Raza, Malik Ahsan Noor Bhutta, Farhat Abbas and Shahid Iqbal. They made my time spent at QAU during my MPhil a wonderful experience. I would like to specially thank my awesome room-mate Naeem Akhtar for providing me excellent environment in our room and for always being a caring friend.

In the end I would like to thank my parents my little sisters and my late phupo, as my postgraduate studies would never have been possible with out their unending love and support.

Tehreem Qasim
February 5, 2015

Abstract

In wireless sensor networks (WSNs), node deployment is one of the most crucial issues as it impacts the cost of network deployment, capability of the network to achieve coverage objectives as well as the life time of WSN. In this thesis, problem of minimum-cost and connectivity guaranteed grid coverage (MCGC) is considered for WSN deployment. Ant colony optimization is an applicable approach for the combinatorial optimization problem of MCGC. A novel ACO based algorithm, ACO-Discreet is proposed in this thesis to obtain solutions for the problem of MCGC. ACO-Discreet works in two phases. In Phase-I, solution building process is carried out like ordinary ACO-based algorithms. Phase-II effectively removes the redundant sensors that are present in the solution obtained from Phase-I and hence considerably reduces the coverage cost. Simulation results have been presented to show the effectiveness of the proposed algorithm.

Contents

Chapter 1

Introduction

Wireless sensor networks have acquired considerable attention recently, especially after advancements in Micro-Electro-Mechanical Systems (MEMS) as well as wireless communications. A WSN is made up of a large number of sensor nodes. These sensor nodes can sense the events they are designed for, process the sensed data and send it to sink via single or multiple hops. Applications of WSNs range from military to civilian, examples include surveillance of hostile territory, agriculture, environmental monitoring etc. Designing a WSN is a difficult task as it affects functioning of the network, cost of its deployment, its capability to sense important events efficiently and its lifetime. It is for this reason that, WSN deployment has become a hot topic for research. Sensor nodes have limited energy as once they are deployed in the field, it is difficult most of the times to replace the battaries when they are depleted. It is, therefore, important to make sure that the network is deployed in such a way that network life time is maximized. In general, the prime objective of

WSN deployment is to achieve coverage and also maintain connectivity of the network. WSN deployment can be divided into two types, i.e deployment based on continued points and discrete grid points. Due to a great deal of advantages, deployment based on discrete grid points has gained attention of researchers recently. In general, a sensor node could be the data originator or serve as relay node to forward another sensor node′s data to the sink. Transmission of sensed data to sink usually takes a many-to-one form. As a consequence of this phenomenon, nodes that are located in close vicinity of the sink have a lot of relay data to forward to the sink and they end up utilizing their battries very quickly. As a result energy hole is created and data can no longer be transmitted to the sink [1]. The problem of energy hole should be considered while designing a WSN. In this thesis the problem of minimum-cost and connectivity guaranteed grid coverage (MCGC) is considered for WSN deployment. The main goal of this problem is to devise an efficient algorithm which can cover all the desired points of interest (PoIs). Deployment cost of WSN deployment is defined in terms of number of sensor nodes placed in the field. Our objective is to minimize the deployment cost in such a way that the functioning of the network is not interrupted and connectivity of the network remains intact. In this thesis a novel ACO based algorithm ACO-Discreet is proposed to solve the problem of MCGC. Care has been taken to maximize the network life time. ACO is an optimization approach in which a complex behavior arises from the interaction of ants. ACO falls in the category of swarm intelligence algorithms and very efficiently provides solutions for combinatorial optimization problems that are NP-hard e.g traveling salesman problem (TSP)[2]. ACO is also suitable for the problem of MCGC. Our main contributions are :

- The heuristic value used ordinarily in ACO algorithms has been modified to place adjacent sensor nodes at larger distances while ensuring connectivity between them.
- The operation of ACO-Discreet is divided into two phases unlike ordinary ACO algorithms which stop after returning a solution. ACO-Discreet continues its operation even after a solution has been built and seeks to reduce the solution further in the Phase-II of the algorithm.

Rest of the thesis is organized as follows. In Ch 2 literature review is presented. In Ch 3 the ACO algorithm has been explained briefly. In Ch 4 three ACO based algorithms Easidesign, ACO-TCAT and ACO-Greedy employed for WSN deployment are discussed. In Ch 5 ACO-Discreet has been proposed. In Ch 6 results and discussion have been presented. In Ch 7 conclusions are given.

Chapter 2

Literature Review

In wireless sensor networks, node deployment is an important issue. It is a fundamental requirement that the sensor network be fully connected with minimum deployment cost. Lot of recent research has focused on designing efficient algorithms which can generate solutions that result in low WSN deployment cost. Reduction in deployment cost is sought by minimizing the number of deployed sensors. Grid based node deployment is presented in [3]. Coverage of k-cover points with minimum number of sensors is NP-complete. In [4] a resource-bounded model is introduced for the optimization of grid coverage problem. In [5] an approach similar to [4] is used and the two proposed algorithms give average coverage as well as coverage to the grid points that are least covered. Simulated annealing is used in [6] to solve the grid based WSN node deployment problem but the effects of position of sink and the connectivity of the network is not considered. In [7] and [8] grid-based placement of sensors is achieved with genetic algorithm, however communication between nodes is not considered. In [9] ant colony optimization is used for WSN deployment but inadequacy of ant transitions leads to substandard solutions. In [10] ACO-TCAT, an ant colony optimization approach with three classes of ant transitions is proposed which results in better solutions compared to [9]. In [11] ACO-Greedy is proposed which uses a greedy migration approach for ants which results in further improvement compared to [10]. In addition non-uniform coverage radii are used in ACO-Greedy to create more node density in areas close to the sink and resolve the problem of energy hole which arises due to more relay burden on nodes that are close to the sink. There are similarities between [9], [10] and [11] but all of them neglect a large number of redundant sensors that are present in the solutions they return. In [12] redundancy check is used but no proper mechanism has been included in the algorithm to remove the redundant sensors. Also the removal of all the redundant sensors could lead to disconnection in the network. ACO-Discreet proposed in this thesis works in two phases. In phase-I a modified heuristic is used to decrease the density of nodes. In phase-II the built solution is further investigated to look for redundant sensors. Thus the unneeded sensors are removed from the solution in such a way that the operation of network is not affected and the coverage cost is considerably reduced.

Chapter 3

Ant Colony Optimization (ACO)

In this chapter ant colony optimization (ACO) is described before we move on to application of ACO algorithm to node deployment in wireless sensor networks.

3.1 Ant Colony Optimization:

Ant Colony Optimization (ACO) comprises artificial systems that take inspiration from foraging behavior of natural ants, that can be used to obtain solutions for discrete optimization problems. Ant colony optimization falls in the category of swarm intelligence algorithms which are inspired by biological systems. The basic theme of swarm intelligence algorithms is that a group of agents or boids interact with each other and with the surrounding environ-

Figure 3.1: Ants Swarm (Image taken from en.wikipedia.org)

ment. The agents abide by a set of rules which ensure interaction among themselves and as a result a global intelligent behavior emerges although there is no central authority to guide them towards that kind of behavior. Besides ACO, other examples of swarm intelligence algorithms include Particle Swarm Optimization (PSO), Artificial Bee Colony Algorithm (ABC), Artificial Immune Systems (AIS), The Grey Wolf Optimizer (GWO) , Stochastic Diffusion Search (SDS) etc.

3.1.1 Stigmergy in Ants:

Stigmergy is the coordination among agents [13]. The effects of an action on the environment provokes future actions and have an impact on them. Hence, consequent actions are based on the previous actions and as a result an ordered and systematic activity takes place.

Stigmergy produces complex and intelligent systems, without any prior plan of action, or central-control [13]. Ants have a very well developed and sophisticated stigmergy.

Natural ants drift in search of food in a random manner, and when they find a food source they return to their colony laying down a chemical called pheromone on the path they traversed. When the other ants come across that path, they are prone to follow the pheromone trail instead of moving around in a random way, and lay down more pheromone on the paths they take while they return from the food source. With the passage of time pheromone laid down on different paths starts to evaporate, and hence its intensity scales down. The longer the path, the longer an ant takes to traverse it. Hence pheromone dimisihes quickly on the longer paths. On the other hand, if a path is short, more ants walk through it and hence more supply of pheromone is laid down before its gets to vaporize and as such large amount of pheromone exists on such paths.

In absence of evaporation the paths that ants chose initially would turn out to be overly appealing to the ants that tread such paths at later stages. This would result in search for the solution being constrained. Thus, the exploration of a short path attracts other ants to that path, and feedback finally steers all the ants to one path. ACO algorithm imitates the behavior of natural ants and constitutes "simulated ants" that roam around the graph in search of a solution to the problem. Example in Figure 3.2 elaborates foraging behaviour of the ants.

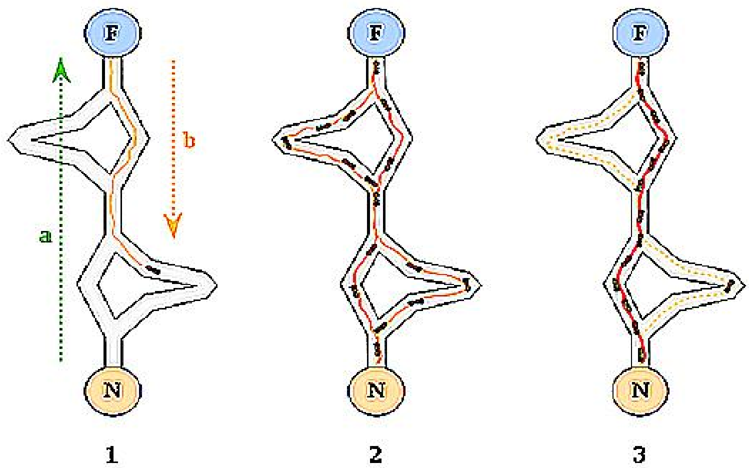


Figure 3.2: Shortest path selection by ants (Image taken from de.wikipedia.org)

- The first ant follows path (a) to get to the food source, and comes back to the nest (N) through path (b) laying down pheromone.
- Initially ants arbitrarily walk through all four possible paths, but gradually the strength of pheromone becomes highest on the shortest path and more ants tend to follow that path.
- Finally all ants converge to the shortest path and the pheromone on the other paths vanishes.

3.1.2 ACO Algorithm:

ACO was proposed in 1991 by Marco Dorigo in his PhD thesis Optimization, learning, and Natural Algorithms, in which he modeled the approach of ants in real world to solve problems. Lot of research has focused on modifications to the original algorithm in recent years.

Ant colony optimization works for a predefined number of iterations. At each iteration, a group of artificial ants build solutions by visiting one vertex after another on a graph. During the tour, they have to follow the condition of picking a particular vertex once only. During the solution building process, at every step, next vertex is selected by an ant with probabilistic decision making. The ant is inclined in favour of a vertex with more pheromone. When an iteration ends, the values of pheromone are modified in accordance with the quality of solutions obtained so as to make the future ants more inclined in favour of the best solutions [14, 15].

ACO can be used to obtain solutions for discrete optimization problems with following characteristics [16].

- $C = \{c_1, c_2, ...c_{N_C}\}$ A finite set of *components*
- $L = \{l_{c_i c_j} | (c_i c_j) \in C\}; |L| \leq N_c^2$ is a finite set of potential *links* among elements of C.
- $J_{c_i c_j} \equiv J(l_{c_i c_j}\ ;\ t)$ is a *cost function* associated with each link $l_{c_i c_j} \in L$

,with possible dependence on time t.

- $\Omega \equiv \Omega(C; L; t)$ is a finite set of *constraints* on elements of sets C and L.
- $s = s < c_i, c_j......, c_k, ... >$ is a sequence over the elements of C or that of L.

  A sequence s is also called a state of the problem. If S is the set of all possible sequences, the set **S** of all the (sub) sequences that are practicable keeping in view the constraints $\Omega(C; L; t)$, is a subset of S. The set **S** defines the feasible states of an optimization problem. The number of elements in a sequence, is denoted by $|s|$.
- Ψ is a solution if it is an element of **S** and fulfills all the requirements of a given problem. A solution may be multi-dimensional in terms of more than one distinct sequences over the elements of C.
- $J_\Psi(L, t)$ is a cost linked to each solution Ψ. $J_\Psi(L, t)$ is a function of all the costs that pertain to the connections belonging to the solution Ψ.

Ants participating in the search for solution possess following properties [16]:

- Each ant searches for a solution with minimum cost $J_\Psi = minJ\ _\Psi(L, t)$.
- Each ant k has memory M_k that's used to store information about the path it has traveled along. Memory can be used for constructing solutions, to

assess the built solutions, and to follow the treaded path in backward direction.

- An ant k in state $s_r = < s_{r-1}, i >$ can move to any node j in its feasible neighborhood N_i^k, defined as $N_i^k = \{j | (j \in N_i) \ \hat{} \ (< s_r, j > \in \mathbf{S})\}$.
- Ant k is given an initial state s_s^k and one or more termination conditions e_k. Typically, the initial state is a sequence of unit length.
- Ants begin from the initial state and keep moving to states in the feasible neighborhood, constructing the solution step by step. The construction of solution ends when at least one of the termination conditions e_k is fulfilled.
- $k - th$ ant on node i can go to a node j selected from N_i^k. The transition is made with a probabilistic decision rule given as.

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}]^\beta}{\sum_{l \in N_i^k} [\tau_{il(t)}]^\alpha [\eta_{il}]^\beta} \qquad j \in N_i^k \tag{3.1}$$

  τ_{ij} and η_{ij} show the pheromone value and heuristic desirability respectively. Parameters α and β control the weightage of pheromone trail and heuristic value.

- After a solution is constructed, the ant can trace the path used in solution in backwards direction and update the pheromone values on the arcs that have been traversed. Pheromone update is given as;

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}(t) \tag{3.2}$$

$\rho \in (0,1)$ is the pheromone decay coefficient which controles the evaporation of previous pheromone value on all edges [17]. New pheromone deposited by ant k is give as;

$$\Delta\tau^{k}(t) = 1/L^{k}(t) \tag{3.3}$$

Hence new deposited pheromone is the reciprocal of the ant's tour length, i.e, shorter tours receive more pheromone and vice versa. Similarly, it is evident from equation 3.2 that the larger the number of ants pass through a particular link, the more pheromone $\Delta\tau^{k}(t)$ is deposited on it.

Hence pheremone value remians high on paths that are short and through which more ants pass.

Chapter 4

ACO algorithms for WSN deployment

In this chapter, three ACO algorithms Easidesign [9], ATO-TCAT [10], and ACO-Greedy [11] have been discussed for node deployment in wireless sensor networks. Coverage requirements of wireless sensor network are formulated as minimum-cost and connectivity guaranteed grid coverage $(MCGC)$ problem. In $(MCGC)$ problem sensors have to be deployed in field composed of discrete grid points. The purpose of sensor deployment is to cover points of interest *(PoIs)* also called critical points. Our goal is to meet the coverage objectives in such a way that each sensor is connected to the sink and minimum number of sensors are deployed.

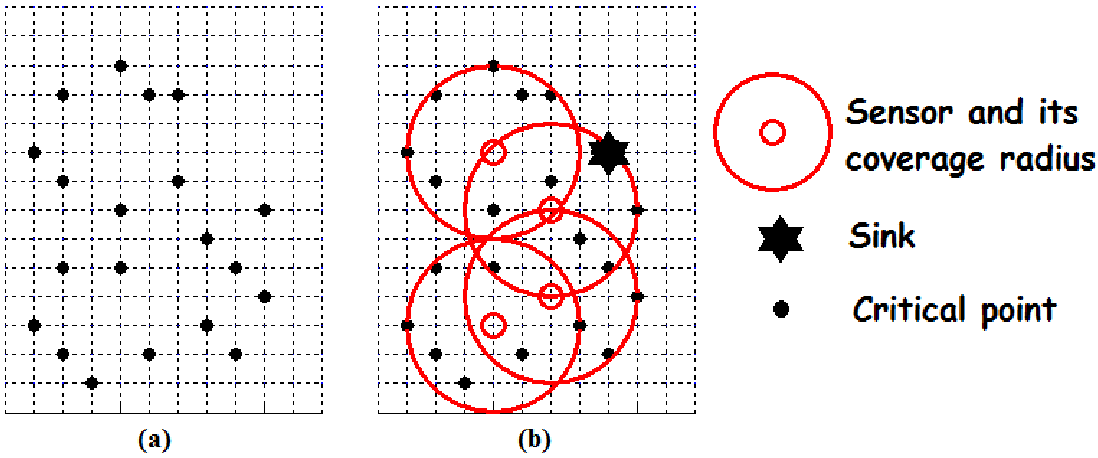


Figure 4.1: MCGC Problem (a) Before sensor deployment (b) After sensor deployment

4.1 EasiDesign:

Easidesign algorithm [9] is described in detail in this section.

4.1.1 Problem Formulation:

The sensing field is composed of discrete points. Sensors are to be deployed on these points. The sensing field is specified as a connected graph $G =< V, E >$. The set V contains the candidate points for sensor nodes deployment. Set E comprises links (u, v), where $u, v \in V$. Communication radius of sensor nodes is denoted by r_c. The set of critical points which have to be covered by sensors is denoted by C, where C is a subset of V. Binary sensor model has been used in which an event can be detected within the sensing radius r_s. For a point v set R_v contains the candidate points which lie in its sensing region. The Boolean variable, $i_v^c = 1$ if $c \in R_v$ and 0 otherwise.

Our goal is to find the smallest set P composed of candidate points where placement of sensors ensures coverage of all critical points. Each point $v \in P$ of the solution should be connected to the sink via a simple path $P_{v,t}$.

4.1.2 Easidesign Algorithm:

In EasiDesign algorithm [9] ants locomote and deposit sensors on the grid points that they travel along in the sensing field. The goal is to cover all the critical points. Points of solution are chosen probabilistically at each step considering the pheromone strength on the links between the ant's current location and the next possible points. When all ants finish building a solution, the solution with smallest number of sensor nodes is cached. This process is repeated till a predetermined number of maximum iterations is completed and in the end the best solution is selected. The input parameters in the pseudocode include:

- k, the least covered times of a critical point.

- c_{ant} , total number of ants in one iteration.
- i_{max} , maximum number of iterations.
- n_e , period of pheromone constraining (explained in Section 3.1.5).

EasiDesign Algorithm:

---

1. EasiDesign(k ,c_{ant}, i_{max}, n_e)
2. Initial point for each ant is selected randomly;
3. $i = 0$; // i shows the iteration number;
4. **WHILE**$(i \leq i_{max})$ **DO**
5.   **FOR** $k = 0$ *to* $k = c_{ant}$ **DO**
6.   SelectNextPoint(j);
7.   // Next point is selected probabilistically;
8.   **END FOR**
9.   Select the solution with minimum cost and enter it in the set P;
10.   GlobalPheromoneUpdate();

IF$(i\%n_e == 0)$ **THEN**

PheromoneConstraining$()$;

END IF

i++;

END WHILE

Select the best solution in set P;

RETURN the best solution;

In EasiDesign [9] there are three major components:

- Function *SelectNextPoint()* to select the next point in a probabilistic way.
- Function *GlobalPheromoneUpdate()* to modify pheromone values after each iteration.
- Function *PheromoneConstraining()* to make adjustments in pheromone values to avert premature convergence.

4.1.3 Next Point Selection:

For next point selection, first off an ant finds out the possible candidate points, then the ant applies a stochastic decision rule to choose one of the candidate points. All points that lie with in communication radius of current point are next possible points. The points on which sensor placement can cover the uncovered critical points are considered first by the ants. The strategy is formulated as follows:

$$Ni^v = \begin{cases} N^v_{efft}, & N^v_{efft} \neq \emptyset \\ N^v_{full}, & N^v_{efft} = \emptyset \end{cases} \tag{4.1}$$

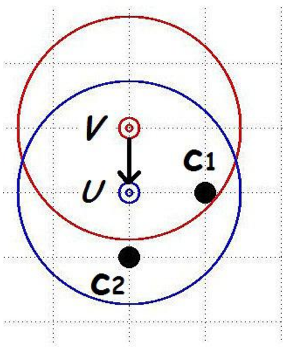
Set N_i^v holds the next possible points for ant i on point v. Set N^v_{full} contains all those points which lie with in the communication radius of the sensor on point v. N^v_{efft} which is a subset of set N^v_{full} includes those points on which, placement of a sensor can cover at least one such critical point that has not been covered before. For an ant i on v, selection probability of a next possible point u is:

$$P^i_{v,u} = \frac{[\tau_{v,u}]^\alpha [\eta^i_{v,u}]^\beta}{\sum\limits_{m \in N_i^v} [\tau_{v,m}]^\alpha [\eta^i_{v,m}]^\beta} \qquad , u \in N_i^v \tag{4.2}$$

Where $\tau_{v,u}$ gives the pheromone value for link (v,u), parameter α controls the impact of $\tau_{v,u}$, parameter β controls the impact of $\eta^i_{v,u}$. $\eta^i_{v,u}$ represents the heuristic desirability of link(v,u).

$$\eta^i_{v,u} = [\sum_{m \in C_u} \gamma^i_m] + 1 \tag{4.3}$$

Set C_u contains the critical points which fall within coverage radius of sensors on point u. γ_m^i shows how many times a critical point m is "uncovered". $\eta_{v,u}^i$ shows how many criticle points a sensor could cover when placed on point u. In Figure 4.2, ant i is on point v and u is a candidate point for next point selection. Placement of a sensor on point u will cover critical points c_1 and c_2. For 2-coverage deployment, c_1 should be covered by one more sensor hence $\gamma_{c_1}^i = 1$. c_2, on the other hand, has not been covered by any sensor so far, therefore,$\gamma_{c_2}^i = 2$. Consequently, the over all desirability of point u is 4. The constant 1 is added to avoid division by zero.



For 2-coverage $\eta_{v,u}^i = 4$

point	covered	r_c^i
c_1	1	1
c_2	0	2

For 3-coverage $\eta_{v,u}^i = 6$

point	covered	r_c^i
c_1	1	2
c_2	0	3

Figure 4.2: Example of the uncovered times γ_m^i and the heuristic desirability $\eta_{v,u}^i$ for $link(v, u)$.

4.1.4 Pheromone Update:

When solutions are constructed by all ants, the pheromone values are updated depending on quality of the solutions that are built [18].

The pheromone for a link is updated as follows:

$$\tau_{v,u} = (1-\rho)\tau_{v,u} + \Delta\tau_{v,u}^{best} \tag{4.4}$$

ρ is the pheromone evaporation coefficient. After all ants build their solutions existing pheromone on all links evaporates by a factor $(1-\rho)$. New pheromone that's deposited is given by $\Delta\tau_{v,u}^{best} = 1/(L_{best}H_{best}^{u})$ if best solution includes the point u in current iteration otherwise $\Delta\tau_{v,u}^{best} = 0$. L_{best} is the number of sensors employed in the best solution achieved. The term H_{best}^{u} prevents solutions with high routing cost. In Figure 4.3 a, sensor on point v has an unreasonable route to the sink. As could be seen in Figure 4.3 b, inclusion of H_{best} in pheromone update process resolves the problem.

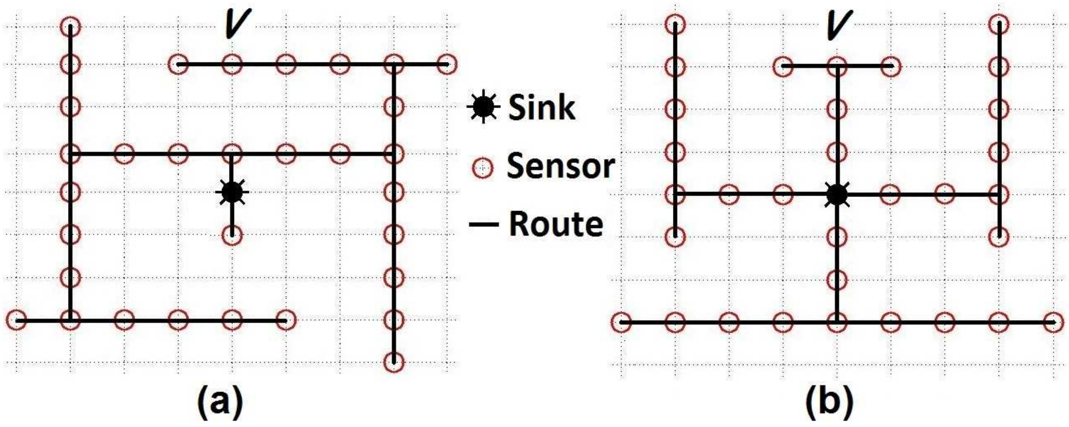


Figure 4.3: Example of route optimization.

4.1.5 Pheromone Constraining:

Pheromone Constraining prevents the value of pheromone from increasing or decreasing beyond given limits, $\tau_{min} \leq \tau_{v,u} \leq \tau_{max}$. τ_{max} is given as under,

$$\tau_{max} = 1/\rho L_{best} \tag{4.5}$$

The process of pheromone constraining dosen't occur after every iteration rather its repeated periodically after a given number of iterations. Variable n_e controles the frequency of occurrence of pheromone constraining process.

4.2 ACO-TCAT:

In ACO-TCAT [10] three classes of ant transitions have been proposed for next point selection. Increase in ant transitions results in better solutions.

4.2.1 Class I of Ant Transition:

ECPs (Effective candidate points):

ECPs are those candidate points on which sensor placement could cover atleast one uncovered critical point.

According to first ant transition, the set of candidate points for next point selection for an ant on point i is;

$$S^{i}_{candidate}(I) = \begin{cases} S_{ECP(i)}, & \text{if } S_{ECP(i)} \neq \phi, \\ \phi & \text{otherwise.} \end{cases} \quad (4.6)$$

The set $S_{ECP(i)}$ contains the *ECPs* that lie with in the communication radius of the sensor on point i.

4.2.2 Class II of Ant Transition:

If there are no *ECPs* with in the communication radius of point i, Class II of ant transition is employed.

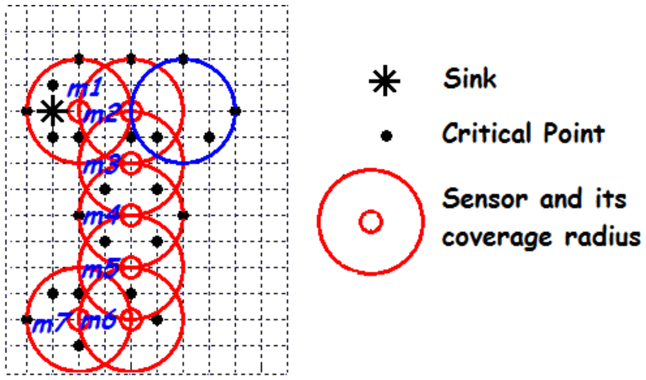


Figure 4.4: Example of ESPs.

ESPs (Effective selected points):

$ESPs$ are those points of solution inside whose communication radius there is atleast one ECP.

According to second ant transition, the set of candidate points for next point selection for an ant on point i is;

$$S^i_{candidate}(II) = \begin{cases} S_{ESP(i)}, & \text{if } S_{ESP(i)} \neq \phi, \\ \phi & \text{otherwise.} \end{cases} \tag{4.7}$$

$S_{ESP(i)}$ contains all the $ESPs$ inside the network. The second ant transition is illustrated in the Figure 4.4. Using the first ant transition, the ant keeps moving from point m_1 to m_7. As it reaches the point m_7, it looks for those points of solution where sensors have already been deployed and $ECPs$ exist with in their communication radii. Point of solution m_2 contains $ECPs$ within its communication radius. Consequently, the ant moves to the point m_3 so that it may move to an ECP in next step to cover uncovered critical points. If at some stage during the search for solution, more than one $ESPs$ exist, either of them could be selected randomly.

4.2.3 Class III of Ant Transition:

In case $S^i_{candidate}(I) = \phi$ and $S^i_{candidate}(II) = \phi$, third class of ant transition is employed.

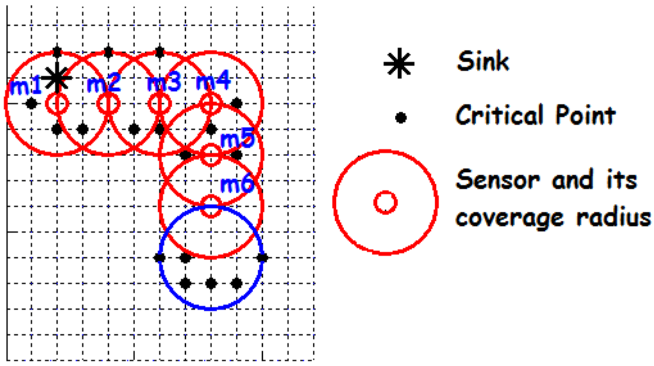


Figure 4.5: Example of RCPs.

RCPs (Residual candidate points):

$RCPs$ are those candidate points on which sensor placement does not cover any critical point.

According to third ant transition, the set of candidate points for next point selection for an ant on point i is;

$$S^{i}_{candidate}(III) = \begin{cases} S_{RCP(i)}, & \text{if } S_{RCP(i)} \neq \phi, \\ \phi & \text{otherwise.} \end{cases} \tag{4.8}$$

$S_{RCP(i)}$ is the set of $RCPs$ within the communication radius of point i. The third ant transition is illustrated in Figure 4.5. The ant keeps moving from point m_1 to point m_5 using the first ant transition, but it runs out of $ECPs$ on point m_5. Third ant transition is then employed and the ant moves from point m_5 to the RCP m_6. $ECPs$ are located within the communication radius of sensor on m_6 and the ant can move to any of them using the first ant transition.

ACO-TCAT Algorithm:

---

1. Initialize all parameters.
2. Each ant moves to a new location by employing one of the three classes of ant transitions.
3. Set of points of solution is updated.
4. Candidate points for ant transition of Class I are updated for every ant.
5. Candidate points for ant transition of Class II are updated for every ant.

6. Candidate points for ant transition of Class III are updated for every ant.
7. Solution built by each ant is checked, in case of incomplete solution, go to step (2).
8. All the constructed solutions are assessed.
9. Pheromone values are updated.
10. If the iterations are not completed, go to step (2).
11. The best solution is selected.

In ACO-TCAT [10], an ant on point i uses the following decision rule to select a particular candidate point, among the potential candidate points, for next sensor deployment.

$$P_{i,j}(t) = \frac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum\limits_{r \in S^{i}_{candidate}} [\tau_{ir}(t)]^{\alpha}[\eta_{ir}(t)]^{\beta}} \tag{4.9}$$

The heuristic desirability $\eta_{i,j}(t)$ is given as;

$$\eta_{ij}(t) = L + \sum_{m \in S_{ECP(j)}} l(m) \tag{4.10}$$

$l(m)$ is a constant 1. Hence $\eta_{ij}(t)$ gives preference to points with more *ECPs* in their vicinities. Constant L is added to avoid division by zero.

The intensity of pehremone on each edge is updated as;

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t) \quad (4.11)$$

Where,

$$\Delta\tau = C/total(t) \quad (4.12)$$

$total(t)$ shows the length of a solution. C is any constant value greater than zero. In ACO-TCAT [10] pheromone constraining process similar to Easidesign [9] is followed.

4.3 ACO-Greedy:

In ACO-Greedy algorithm [11], a greedy migration scheme is used for movement of ants. Furthermore, non-uniform sensing/communication radii are used to resolve energy hole problem which arises due to more relay traffic burden on nodes close to the sink, as a result these nodes end up depleting their batteries very quickly. In ACO-Greedy algorithm[11], the nodes close to the sink are assigned small communication radii. The communication radii of nodes keep on increasing as the distance from sink increases. Hence node density is greater near the sink, which helps controle energy hole problem. The communication radius for a node i is given as;

$$R_i = [1 - \frac{d_{max} - d(i, sink)}{\mu(d_{max} - d_{min})}]R_{max} \quad (4.13)$$

d_{max} and d_{min} show maximum and minimum distances between the sink and sensor nodes. μ controls minimum communication radius e.g setting $\mu = 2$, gives values of communication radii between $R_{max}/2$ and R_{max} depending on distance of sensor node from the sink.

4.3.1 ACO-Greedy Algorithm:

Definition 1. (ECP):

ECPs (Effective Candidate Points) are those candidate points on which sensor placement could cover atleast one uncovered critical point.

Definition 2. (PoO):

$PoO(Point\ of\ Origination)$ is defined as a point from which an ant moves to another point after stochastic decision making.

e.g, If an ant moves from point i to point j, point i is a PoO.

Definition 3. (PoD):

PoD (Point of Destination) is defined as a point to which an ant comes from another point after stochastic decision making.

e.g, if an ant moves from point i to point j, point j is a PoD. In the ACO algorithms, generally the latest *PoD* is the next *PoO*. This could result in increased coverage cost if there are no *ECPs* within the coverage radius of the latest *PoD*. Similarly, even the presence of one or more *ECPs* within the coverage radius of latest *PoD* may not give best results if the said *ECPs* cover less critical points compared to other *ECPs* present in the network. As a result, many sensors present in the network cover less critical points. Ant's greedy migration addresses this problem by making prudent selection of next *PoO*.

Definition 4. (OPOs):

The points among the *PoSs (Points of solution)* that have one or more *ECPs* within their coverage radius are called *OPOs* (Ordinary Points of Origination).

Definition 5. (SPO):

The point among the *PoSs (Points of solution)* with the largest number of *ECPs* with in its coverage radius is called *SPO (Superior Point of Origination)*.

In ACO-Greedy [11], instead of the latest *PoD*, the *SPO* is chosen as the next *PoO*. If more than one *SPOs* exist, any of them can be chosen randomly as next *PoO*. Since an *SPO* has more *ECPs* with in its coverage radius, migration to an *SPO* ensures that more critical points are covered at each ant's step and in this way coverage cost is reduced.

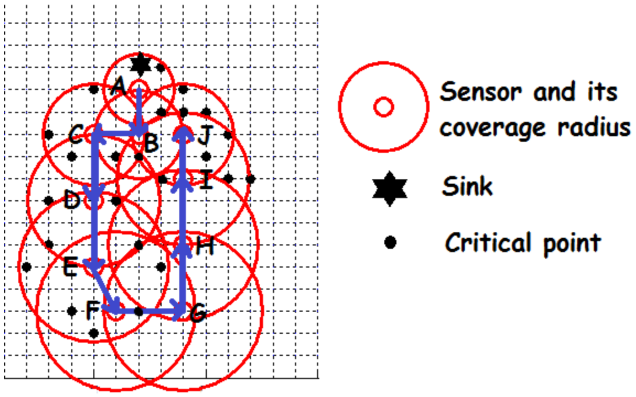


Figure 4.6: Sensor deployment with ordinary ACO.

In Figure 4.6, the solution obtained using ordinary ACO is $\{sink, A, B, C, D, E, F, G, H, I, J\}$. 10 sensors fulfill coverage requirements in this case.

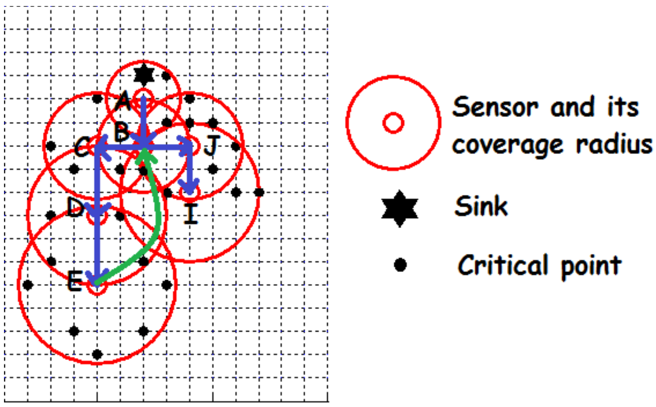


Figure 4.7: Sensor deployment with ACO-Greedy.

In ACO-Greedy [11] when the ant reaches point E (Figure 4.7), it dosen't find a lot of *ECPs* there, as such point E is not considered as the next *PoO*. Instead it migrates to point B which is an *SPO*. Hence point B is selected as the next *PoO*. The ant moves from point B to point J and I. The solution obtained is given as:

$\{sink, A, B, C, D, E, I, J\}$. This time, the same coverage objectives have been achieved with 7 sensors.

ACO-Greedy Algorithm:

Initialize all parameters.

while(the iterations are not completed) **do**

for(each ant) **do**

while(all the points of interest are not covered) **do**

for(each PoS) **do**

Find out current SPOs;

end for

if(the current PoD is not an SPO) **then**

migrate to an SPO chosen randomly and make it PoO;

end if

calculate the transition probability of each candidate point;

move to one of the next possible points following formula (4.9);
set comprising points of solution is updated;
set that holds PoIs that haven't been covered is updated;
set containing ECPs is updated;
end while
end for
assess each solution;
choose best solution;
update the pheromone values;
end while
return the best solution;

In ACO-Greedy [11], probability for selection of next point $P_{ij}(t)$, $\eta_{i,j}(t)$, and pheromone update $\tau(t+1)$ are same as described in ACO-TCAT [10] in Equations 4.9, 4.10 and 4.11 respectively. Pheromone constraining process similar to ACO-TCAT [10] is also employed to make sure that values of pheromone stay with in limits.

Chapter 5

ACO-Discreet algorithm

In the previous chapter three algorithms were discussed for WSN deployment to meet the requirements of *MCGC* problem. In this chapter a new algorithm ACO-Discreet is proposed that overcomes deficiencies in the aforementioned three algorithms to further reduce the coverage cost and achieve better solutions for *MCGC* problem. In phase-I of ACO-Discreet, non-uniform coverage radii for sensor nodes and greedy migration scheme similar to [11] have been used. Also *ECPs* are defined as those points on which sensor placement can cover one or more uncovered points of interest (PoIs) [11]. For a sensor on point i feasible neighbourhood contains all those next possible grid points where the next probable sensor′s sensing radius overlaps with that of the sensor on point i. To increase the distance between two adjacent sensors, a modified heuristic value is used which is not only dependent on the number of

ECPs with in the coverage radius of the sensor on point *j* but also the distance of sensor on point *j* from the one on point *i*. The heuristic value is given as,

$$\eta_{ij}(t) = L + \sum_{m \in S_{ECP(j)}} l(m) + Dist_{ij} \tag{5.1}$$

When Phase-I of the algorithm returns a solution, Phase-II of the algorithm starts its operation and seeks to remove all the sensors that are redundant in nature. Phase-II comprises tours conducted by two ants only.

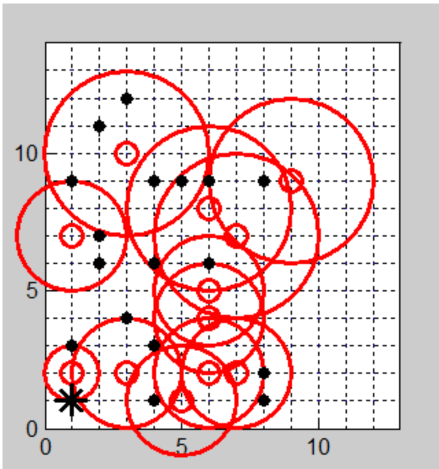


Figure 5.1: Solution obtained at the end of Phase-I of ACO-Discreet.

The 1st ant conducts its tour on the solution obtained from Phase-I and builds a parallel solution of its own. It starts off from the last sensor in the solution and moves in the backwards direction and checks for all those sensors that have *PoIs* in their coverage radii covered more than the desired number of times. If such a sensor is found whose removal doesn't render any critical point uncovered, it's removed from the solution.

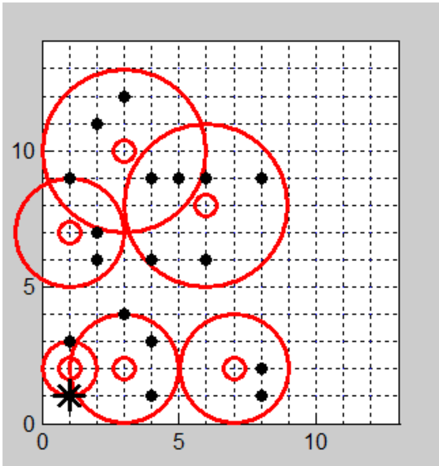


Figure 5.2: Solution built by 1st ant in phase-II

At the end of the tour of 1st ant, a reduced solution is obtained with all the unwanted sensors gone. But some times the connectivity of the network is disrupted as could be observed from Fig 5.2. To resolve this problem a 2nd ant conducts its tour on the solution of the first ant to look for a possible disconnection in the network and restore network connectivity. Unlike the 1st ant that executed its operation considering the sensing radii, the 2nd ant is concerned with the communication radii of sensor nodes. It starts off from the sink and during the first step of the tour, looks for all those sensors in the solution of first ant that are connected to the sink via single hop and adds those points to its solution. At the second step of its tour, it looks for all those sensors in the solution of 1st ant that are connected to the solution it has built so far and so on. The 2nd ant keeps on adding sensors from the solution of the first ant to its own until it reaches a point where one or more sensors in the solution of the first ant are not connected to the solution of its own.

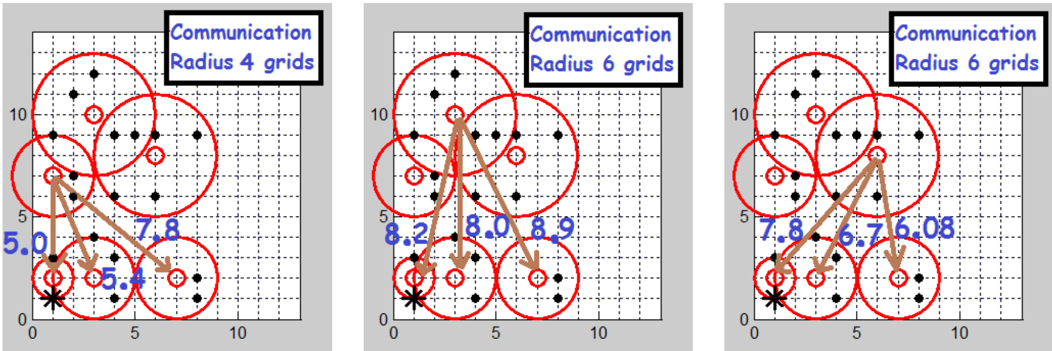


Figure 5.3: Disconnection found.

Thus when the disconnection is identified it looks for all the senors in the original solution obtained at the end of Phase-I, that are connected to the disconnected portion of the network in the 1st ants solution and saves them in a temporary data structure. Then it looks for all those points in the original solution that are connected to its own solution and saves them in another temporary data structure.

The common sensors between the two temporary data structures are the ones could be used to restore network connectivity. Hence, one of these common sensors is reinstated as shown in Fig. 5.4. At the end of Phase-II a large number of the unwanted sensors are removed in such a way that the functioning of the network remains unaffected.

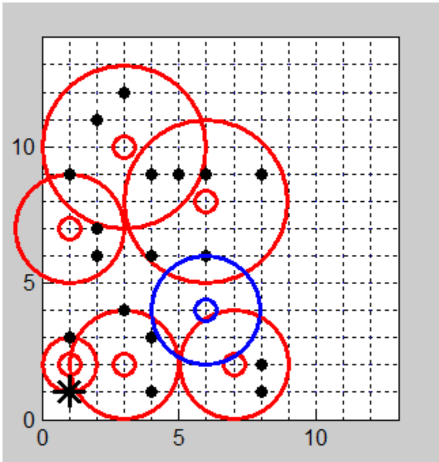


Figure 5.4: Solution built by 2nd ant in phase-II

In ACO-Discreet probability for selection of next point $P_{ij}(t)$ and pheromone update $\tau(t+1)$ are same as described in in Equations 4.9 and 4.11 respectively. Pheromone constraining process similar to previously described algorithms is also employed to make sure that values of pheromone stay with in limits. Pseudocode is given as under.

ACO-Discreet Algorithm:

Phase-I

Initialize all parameters;

WHILE(Maximum number of iterations Imax have not been reached)DO

Each ant builds its solution;

Cache the best solution for current iteration;

Update the pheromone values;

END WHILE

Select the minimum cost solution;

Phase-II

Execute backwards tour for 1st ant and build a new solution;

IF(A sensor is found with all PoIs in its coverage radius covered more than the required number of times)

Exclude this sensor from the new solution;

END IF

Execute tour for 2nd ant;

IF(Disconnection in network is detected)

Reinstate a sensor from the original solution obtained at the end of Phase-I to restore network connectivity;

END IF

Save the solution;

Chapter 6

Results and discussion

In this chapter, results and discussion a presented. In Chapter 4 three algorithms EasiDesign [9], ACO-TCAT [10] and ACO-Greedy [11] were discussed. In chapter 5 a novel algorithm ACO-Discreet was proposed. In this chapter performance of all four algorithms is evaluated on different scales of network.

6.1 Experimental Setup:

The experiments to evaluate the performance of all four algorithms in terms of node deployment cost and network lifetime are executed in Matlab. Network with 9x9 grid points is used. Sensing radius for all nodes is equal to 2 grids in EasiDesign [9] and ACO-TCAT [10] while in ACO-Greedy [11] and ACO-Discreet, it keeps varying between 1 and 3 grids depending on distance of sensor from the sink node. $\alpha = 1$, $\beta = 2$ and $T_c = 10$ have been used. T_c is the pheromone constraining period. Total 100 iterations are executed and 10 ants participate in the search for solution in each iteration.

6.2 Deployment Cost Comparision:

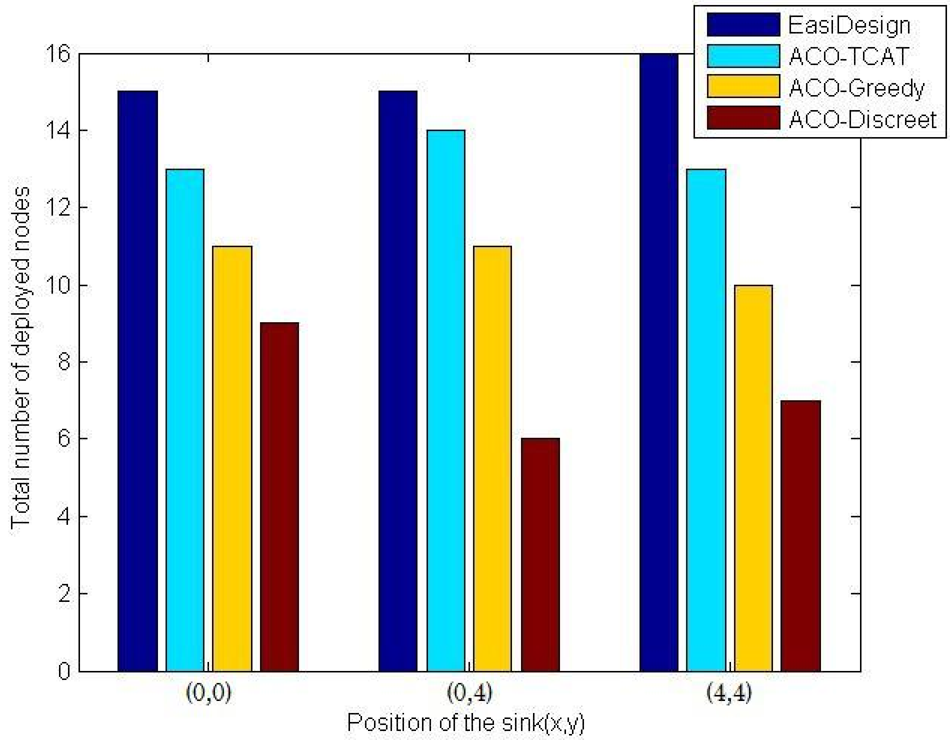


Figure 6.1: 9x9 grid points 20 PoIs.

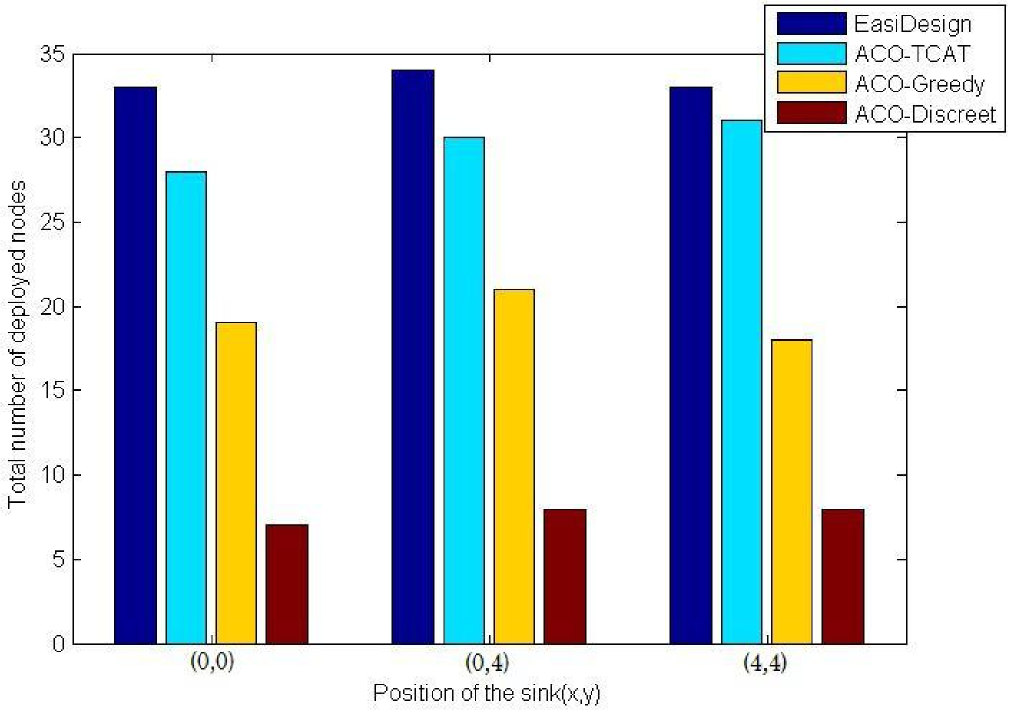


Figure 6.2: 9x9 grid points 40 PoIs.

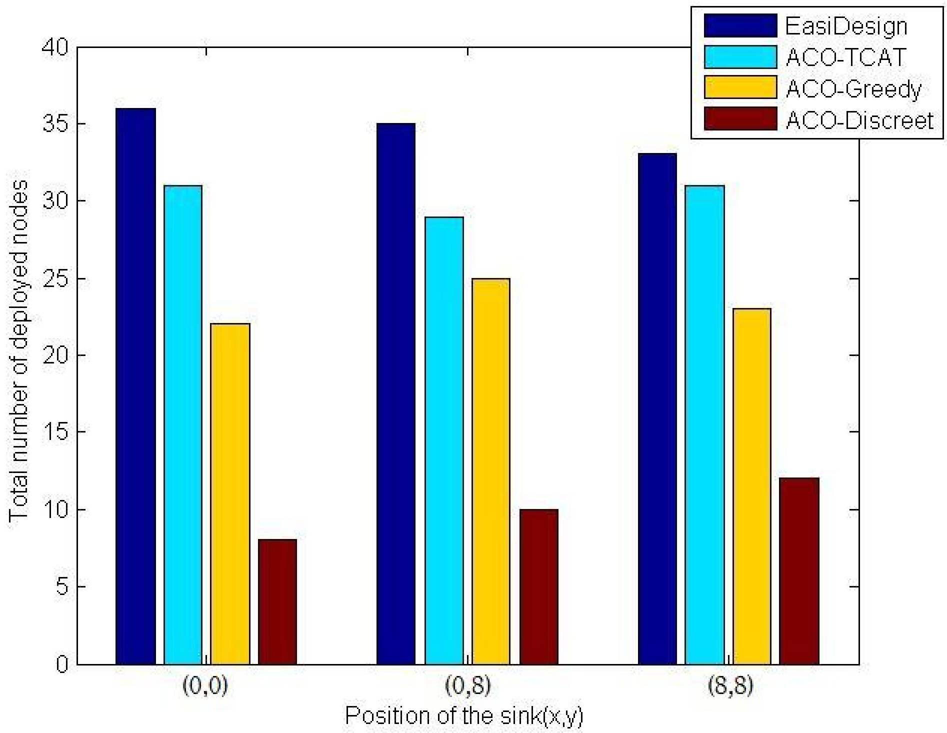


Figure 6.3: 9x9 grid points 40 PoIs.

It could be observed from the simulation results of all four algorithms, that EasiDesign [9] provides solutions with highest number of deployed nodes. ACO-TCAT [10] reduces the deployment cost by utilizing more ant transitions. ACO-Greedy [11] further reduces the deployment cost by employing a greedy migration scheme for next point selection. Of all four algorithms, the proposed algorithm ACO-Discreet provides solutions with minimum deployment cost. This is due the modified heuristic employed for reducing node density in Phase-I of the algorithm and in turn a drastic reduction in the number of deployed nodes in the phase-II which has the potential to significantly reduce the solutions constructed by ordinary ACO algorithms.

6.3 Network Lifetime Comparison:

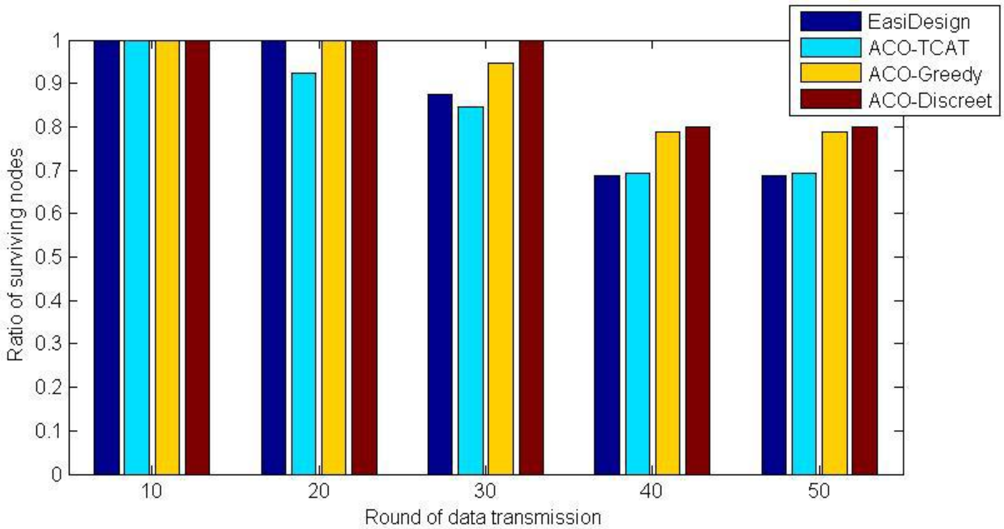


Figure 6.4: 9x9 grid points 20 PoIs.

In the Figure 6.4 life times of networks obtained from different algorithms have been compared. The energy model from [19] has been used. To transmit an $l - bit$ packet, the energy consumed is given as under,

$$E_{Tx}(l,d) = \begin{cases} lE_{elec} + l\epsilon_{fs}d^2, & d < d_o \\ \\ lE_{elec} + l\epsilon_{amp}d^4, & d \geq d_o \end{cases} \tag{6.1}$$

Similarly to receive an l-bit packet, the energy consumed is given as,

$$E_{Rx}(l) = lE_{elec} \tag{6.2}$$

It can be observed that ACO-Greedy [11] performs better than EasiDesign [9] and ACO-TCAT [10] by enhancing the network life time with non-uniform sensor node coverage radii which results in larger node density in areas close to the sink where relay burden is more compared to the areas farther away from sink. In ACO-Discreet similar non-unifrom radii are used at different locations. ACO-Discreet performs slightly better than ACO-Greedy [11] because with reduced number of sensor nodes, there is less relay traffic for nodes close to the sink.

Chapter 7

Conclusions

In this thesis a novel ACO based algorithm ACO-Discreet is proposed to solve the problem of *MCGC* in Wireless Sensor Networks. The objective is to cover all the points of interest with minimum possible number of sensors in such a way that connectivity of the network is guaranteed. The proposed algorithm works in two phases. Phase-I is similar to ordinary ACO based algrithms used to find solution for the problem of *MCGC*. In Phase-I, however, a modified heuristic is used to obtain solutions better than those of other ACO algorithms. In Phase-II further reduction is carried out in the solution obtained from phase-I and a large number of redundant sensors are removed. As a result a very reduced final solution is obtained. The proposed algorithm yields solutions with much less deployment cost compared to other algorithms.

Also network lifetime of solutions obtained from ACO-Discreet is found to be longer than that of other ACO based algorithms.

Bibliography

[1] Cheng Tien Ee and Ruzena Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 148–161. ACM, 2004.

[2] Marco Dorigo and Luca Maria Gambardella. Ant colony system: a cooperative learning approach to the traveling salesman problem. *Evolutionary Computation, IEEE Transactions on*, 1(1):53–66, 1997.

[3] Wei-Chieh Ke, Bing-Hong Liu, and Ming-Jer Tsai. Constructing a wireless sensor network to fully cover critical grids by deploying minimum sensors on grid points is np-complete. *IEEE Trans. Computers*, 56(5):710–715, 2007.

[4] Santpal S Dhillon, Krishnendu Chakrabarty, and SS Iyengar. Sensor placement for grid coverage under imprecise detections. In *Information Fusion,*

2002. Proceedings of the Fifth International Conference on, volume 2, pages 1581–1587. IEEE, 2002.

[5] Santpal Singh Dhillon and Krishnendu Chakrabarty. *Sensor placement for effective coverage and surveillance in distributed sensor networks*, volume 3. IEEE, 2003.

[6] Frank YS Lin and Pei-Ling Chiu. A near-optimal sensor placement algorithm to achieve complete coverage-discrimination in sensor networks. *Communications Letters, IEEE*, 9(1):43–45, 2005.

[7] D Jourdan and Olivier L de Weck. Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 5, pages 2466–2470. IEEE, 2004.

[8] Yong Xu and Xin Yao. A ga approach to the optimal placement of sensors in wireless sensor networks with obstacles and preferences. In *Consumer Communications and Networking Conference, 2006. CCNC 2006. 3rd IEEE*, volume 1, pages 127–131. IEEE, 2006.

[9] Dong Li, Wei Liu, and Li Cui. Easidesign: an improved ant colony algorithm for sensor deployment in real sensor network system. In *Global*

Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, pages 1–5. IEEE, 2010.

[10] Xuxun Liu. Sensor deployment of wireless sensor networks based on ant colony optimization with three classes of ant transitions. *Communications Letters, IEEE*, 16(10):1604–1607, 2012.

[11] Xuxun Liu and Desi He. Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks. *Journal of Network and Computer Applications*, 39:310–318, 2014.

[12] S Singh, S Chand, R Kumar, and B Kumar. Optimal sensor deployment for wsns in grid environment. *Electronics Letters*, 49(16):1040–1041, 2013.

[13] Leslie Marsh and Christian Onof. Stigmergic epistemology, stigmergic cognition. *Cognitive Systems Research*, 9(12):136 – 149, 2008. Perspectives on Social Cognition.

[14] Marco Dorigo, Gianni Caro, and Luca Gambardella. Ant algorithms for discrete optimization. *Artificial life*, 5(2):137–172, 1999.

[15] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *Computational Intelligence Magazine, IEEE*, 1(4):28–39, 2006.

[16] Saad Ghaleb Yaseen and Nada MA AL-Slamy. Ant colony optimization. *IJCSNS*, 8(6):351, 2008.

[17] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29–41, Feb 1996.

[18] Thomas Stützle and Holger H Hoos. Max–min ant system. *Future generation computer systems*, 16(8):889–914, 2000.

[19] Wendi B Heinzelman, Anantha P Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *Wireless Communications, IEEE Transactions on*, 1(4):660–670, 2002.